Morten Dæhlen M
Tom Lyche Jean-L
Knut Mørken Lar

Mathemati for Curves

LNCS 5862

7th International Conferen
Tønsberg, Norway, June/Ju
Revised Selected Papers

Lecture Notes in Computer Science 5862

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Morten Dæhlen Michael Floater
Tom Lyche Jean-Louis Merrien
Knut Mørken Larry L. Schumaker (Eds.)

Mathematical Methods for Curves and Surfaces

7th International Conference, MMCS 2008
Tønsberg, Norway, June 26 – July 1, 2008
Revised Selected Papers

Springer

Volume Editors

Morten Dæhlen
University of Oslo, Norway
E-mail: mortend@ifi.uio.no

Michael Floater
University of Oslo, Norway
E-mail: michaelf@ifi.uio.no

Tom Lyche
University of Oslo, Norway
E-mail: tom@ifi.uio.no

Jean-Louis Merrien
INSA de Rennes, France
E-mail: jean-louis.merrien@insa-rennes.fr

Knut Mørken
University of Oslo, Norway
E-mail: knutm@ifi.uio.no

Larry L. Schumaker
Vanderbilt University, Nashville, TN, USA
E-mail: larry.schumaker@vanderbilt.edu

Library of Congress Control Number: 2010921005

CR Subject Classification (1998): I.3.5, I.4.8, G.1.1-2, F.2.2, G.2, J.6

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-642-11619-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-11619-3 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12833740 06/3180 5 4 3 2 1 0

Preface

The Seventh International Conference on Mathematical Methods for Curves and Surfaces took place June 26 – July 1, 2008, in Tønsberg, Norway. The earlier conferences in the series took place in Oslo (1988), Biri (1991), Ulvik (1994), Lillehammer (1997), Oslo (2000), and Tromsø (2004). The conference gathered 165 participants from almost 30 countries who presented a total of 129 talks. This includes nine invited talks and seven mini-symposia.

This book contains 28 original articles based on talks presented at the conference. The topics range from mathematical analysis of various methods to practical implementation on modern graphics processing units. The papers reflect the newest developments in these fields and also point to the latest literature. The papers have been subject to the usual peer review process, and we thank both the authors and the reviewers for their hard work and helpful collaboration.

We wish to thank those who have supported and helped organize the conference. First and foremost it is a pleasure to acknowledge the generous financial support from the Department of Informatics and the Centre of Mathematics for Applications (CMA) at the University of Oslo, and the Research Council of Norway. We would also like to thank Andrew McMurry for his help with with technical matters, and Sara Mørken for help with the registration.

November 2009

The editors

Organization

Organizing Commitee and Editors

Morten Dæhlen	University of Oslo, Norway
Michael Floater	University of Oslo, Norway
Tom Lyche	University of Oslo, Norway
Jean-Louis Merrien	INSA de Rennes, France
Knut Mørken	University of Oslo, Norway
Larry L. Schumaker	Vanderbilt University, USA

Invited Speakers

Jean-Daniel Boissonnat, Sophia Antipolis, France
Massimo Fornasier, Linz, Austria
Tom Hughes, Austin, USA
Jörg Peters, Gainesville, USA
Ragni Piene, Oslo, Norway
Robert Schaback, Göttingen, Germany
Peter Schröder, Caltech, USA
Jonathan Shewchuk, Berkeley, USA
Joachim Weickert, Saarland, Germany

Mini-Symposia Organizers

Oleg Davydov, Glasgow, UK
Tor Dokken, Oslo, Norway
Bin Han, Edmonton, Canada
Chuck Hansen, Salt Lake City, USA
Rimvydas Krasauskas, Vilnius, Lithuania
Trond Kvamsdal, Trondheim, Norway
Carla Manni, Rome, Italy

Sponsoring Institutions

Department of Informatics, University of Oslo
Centre of Mathematics for Applications, University of Oslo
Research Council of Norway

Table of Contents

MMCS 2008

Partial Differential Equations for Interpolation and Compression of Surfaces

Egil Bae[1] and Joachim Weickert[2]

[1] Department of Mathematics, University of Bergen, Norway
Egil.Bae@math.uib.no
[2] Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science
Building E1.1, Saarland University, 66041 Saarbrücken, Germany
weickert@mia.uni-saarland.de

Abstract. Partial differential equations (PDEs) have recently shown to be very promising for image interpolation and compression. Inspired from this work, we present a PDE based approach to interpolation of surfaces from scattered point sets using the geometric diffusion equation. Triangulated surfaces are considered in the discrete setting, and the geometric diffusion equation is discretized by the finite element method directly on the triangular mesh. Furthermore, a PDE based method for lossy compression of triangulated surfaces is presented. The idea is to store only a few relevant vertex coordinates in the encoding step. In the decoding step, the remaining vertices are reconstructed by solving the geometric diffusion equation. Finally, two modified reconstruction methods are proposed that are shown to improve the compression quality for both images and surfaces. These reconstruction methods approximate instead of interpolating, and have links to Hopscotch methods for the numerical solution of PDEs. Experiments are presented illustrating that results of high quality can be obtained using simple geometric diffusion without any information on surface normals.

1 Introduction

Compression of mesh geometry is getting increasingly more important. Surfaces of millions, and even billions of vertices can get acquired. In order to handle such large amount of data, compression is necessary.

Until recently, the most competitive image and surface compression methods are based on decorrelating the data in the frequency domain, using methods such as wavelets or the cosine transform [1,2,3]. For image compression, the work of [4] proposed a very promosing alternative method, based on interpolation in the spatial domain with PDEs.

Inspired from this work, we propose a PDE based method for surface interpolation from sparse scattered point sets, and a PDE based method for lossy compression of triangulated surfaces. The interpolation method is formulated in the continuous setting as finding the steady state of a time dependent geometric PDE. In the discrete setting, triangulated surfaces are considered, and the PDE is discretized by the finite element method, yielding a discrete flow for each vertex coordinate.

The compression method encodes the surface by storing only a suitable small subset of the vertices. The surface is reconstructed in the decoding step by interpolating from the scattered set of points by solving a geometric PDE.

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 1–14, 2010.

In this work, we particularly study the simple geometric diffusion equation, and show that it has surprisingly good interpolation properties when the interpolation points are selected properly. Furthermore, an improvement is presented based on this equation, which links approximation and numerical methods for PDEs. A similar improvement is also presented for image compression, showing that the linear diffusion equation can yield equally good results as more complicated nonlinear PDEs.

The goal of this paper is to prove the concept of the new surface compression method. We aim to make comparisons with other methods at later stage of development. At the current stage it is hard to assess the exact compression rate of our method. Also, in contrary to wavelets which require semi-regular meshes, our method is applied directly to irregular meshes. This makes a direct comparison difficult.

The structure of the paper is as follows: In Section 2 we describe the continuous framework behind our surface interpolation. A finite element discretization is presented in Section 3. In the fourth section we give details on our baseline method for surface compression, while in Section 5 we introduce two modifications that avoid singularities at the interpolation points. Numerical experiments are performed in Section 6, and the paper is concluded with a summary in Section 7.

Relation to previous work. PDEs have been used for a growing number of image and surface processing problems in the recent years. They have shown to be very powerful for problems such as inpainting, where one wants to fill in regions of missing information using information at the boundary of the regions [5, 6, 7, 8, 9]. Interpolation is a special type of inpainting problem, where the data is only known on a scattered set of points. Until recently, PDE based methods have been little studied for this problem.

For surface interpolation, one method based on solving the Poisson problem was proposed in [10]. Contrary to this work, we don't need any information about the normal vectors. From a compression point of view this is an advantage, since normal vectors are expensive to store. For unoriented dense point clouds, a level set approach was proposed in [11], based on finding the surface minimizing a functional depending on the distance to the interpolation points. In [12, 13], Kobbelt et. al. proposed a method for interpolating a fine polygon from the vertices of a coarse polygon by minimizing a function of linearized curvature.

For image interpolation, the recent work of [4] demonstrated that certain PDEs are very promising. Especially edge enhancing anisotropic diffusion [14], where an anisotropic diffusion tensor suppresses diffusion across edges and allows diffusion along them, performed well. Also the linear diffusion equation yielded good results, provided that the interpolation points were chosen carefully. The image was reconstructed by solving a Dirichlet boundary value problem with the interpolation data as boundary conditions. Furthermore, a new image compression method was developed based on these results, shown to outperform the well-known JPEG standard at high compression rates. The idea was to store only pixels that were hard to reconstruct by the PDE in the encoding step. In the decoding step, the image was interpolated from this sparse set of pixels. Selection of the best interpolation points is a difficult combinatorial optimization problem. For the linear diffusion equation, [15] showed that the exact solution could be computed. For other PDEs, suboptimal solutions could be estimated by B-tree triangular coding, or stochastic approaches [4].

We wish to extend the image interpolation ideas to surfaces. Locally, the coordinate maps of a surface are very similar to an image. We therefore formulate PDEs, such that the differential operators act directly on the coordinate maps. In this sense, the geometric diffusion equation can be seen as the surface equivalent of the linear diffusion equation.

The geometric diffusion equation and the Laplace-Beltrami operator have been extensively used in geometric modelling. Typically for the purpose of surface fairing or smoothing [16, 17, 18, 19, 20]. In [21], a mesh encoding technique is developed based on quantization of the discrete Laplace-Beltrami operator applied to the mesh.

2 PDE Based Surface Interpolation

In this section, we present the continuous formulation for our PDE based method for surface interpolation from sparse scattered point sets.

We begin with some basic notation. Let S denote a smooth, compact, orientable surface embedded in $\mathbb{R}^3$. S can be parameterized by $\{x^\alpha, \Omega^\alpha\}_\alpha$, where $x^\alpha(\xi_1^\alpha, \xi_2^\alpha) : \Omega_\alpha \mapsto S$ are the coordinate maps (patches). For ease of notation, we from now on drop the subscript α. We let ∇_S denote the intrinsic gradient operator of S. Applied to a function $u \in \mathscr{C}^1(S)$, this operator can be written in local coordinates

$$\nabla_S u = \sum_{i,j} {g^{-1}}_{ij} \frac{\partial (u \circ x)}{\partial \xi_j} \frac{\partial}{\partial \xi_i}, \tag{1}$$

where $g_{ij} = I(\frac{\partial x}{\partial \xi_i}, \frac{\partial x}{\partial \xi_j}) = \frac{\partial x}{\partial \xi_i} \cdot \frac{\partial x}{\partial \xi_j}$ are the metric coefficients with respect to x. $I(\cdot,\cdot)$ is the first fundamental form, which at each $p \in S$ is the inner product on the tangent plane T_pS. The Laplace-Beltrami operator Δ_S is a generalization of the Laplace operator for manifolds. Applied to a function $u \in C^2(S)$, it is defined by the duality

$$\int_S \Delta_S u \, \phi \, dx = -\int_S I(\nabla_S u, \nabla_S \phi) + \int_{\partial S} \partial_{n^{co}} u \, \phi \, d\sigma, \quad \forall \phi \in C^\infty(S), \tag{2}$$

where n^{co} is the co-normal, which at each $p \in \partial S$ is the normal vector of the curve ∂S lying in the tangent plane T_pS. Linear diffusion can now be formulated for a function u defined on a surface

$$\frac{\partial u}{\partial t} = \Delta_S u \quad \text{on} \quad S \times [0, \infty) \tag{3}$$

Our interpolation method is based on applying these differential operators directly to the coordinate maps x. We will particularly focus on the geometric diffusion equation, defined as

$$\frac{\partial x}{\partial t} = \Delta_{S(t)} x \quad \text{on} \quad S(t) \times [0, \infty). \tag{4}$$

Note that this equation is non-linear, as the differential operators depend on the unknown x. It can be shown that this equation describes motion by mean curvature (see [22] page 151). A disadvangate of this equation is certain shrinking effects on the volume enclosed by the surface [23]. Incorporation of boundary conditions will counteract most of these shrinking effects. An even better countermeasure is presented in Section 5.

Our interpolation method can now be formulated. We let x^{known} denote the coordinates of the scattered set of points. The surface S is reconstructed by solving the geometric diffusion equation with x^{known} as boundary conditions.

$$\frac{\partial x}{\partial t} = \Delta_{S(t)} x \qquad \text{on} \quad S(t) \times [0,\infty) \tag{5}$$

$$x = x^{known} \qquad \text{on} \quad \partial S(t) \times [0,\infty), \tag{6}$$

$$S(0) = S_0 \tag{7}$$

where S_0 is some initial guess. In order to be well posed in the continuous setting, x^{known} is assumed to be patches of non-zero measure. In the discrete setting, points always have a non-zero measure, so we avoid this problem.

As previously stated, the geometric diffusion equation can locally be seen as the surface equivalent of the linear diffusion equation, since the Laplace-Beltrami operator is applied to the coordinate maps. A different philosophy is to process the normal vectors, which yields higher order PDEs. See [24] for total curvature minimization in the level set framework. See [25] for minimization of Willmore energy and approximations on discrete meshes.

It could interesting to formulate the interpolation method with higher PDEs, however in this paper we focus on second order for a number of reasons: Higher order PDEs requires information of the normal vectors at the boundary in order to have a unique solution. From a compression point of view, storage of normal vectors at the interpolation points would require twice as much space. Numerical solution of higher order PDEs is computationally harder. Finally, we expect oscillation problems, since the boundary is so sparsely scattered. On the order hand, higher order PDEs have many attractive properties such as less shrinking effects, preservation of edges etc. They should be explored for the interpolation problem in the future. A generalization we have strongest belief in is anisotropic geometric diffusion [19].

3 Finite Element Discretization

Discretization of our continuous surface interpolation model is done by the finite element method. Using linear elements results in linear coordinate patches, which suits very well with triangulated surfaces. A finite element discretization for the Laplace-Beltrami operator was first proposed in [18], and for the geometric diffusion equation in [17]. We express equation (5) - (7) in weak form. That is, find $x(t)$ such that

$$\int_{S(t)} \frac{\partial x}{\partial t} \phi \, dx = -\int_{S(t)} I(\nabla_{M(t)} x, \nabla_{S(t)} \phi) \, dx \qquad t > 0 \tag{8}$$

$$S(0) = S_0, \tag{9}$$

for all $\phi \in C^\infty(S(t))^3 = V^3$ with $\phi = 0$ on $\partial S(t)$. Note that the boundary term in (2) vanishes, because the boundary is held fixed.

Spatial discretization is made by reducing V to a finite dimensional subspace $V_h \subset V$. The surface coordinate maps are thus reduced to a finite dimensional subspace. This will result in a discrete surface S_h. Letting $X(t)$ denote the discrete coordinate map, we then obtain

$$\int_{S_h(t)} \frac{\partial X}{\partial t} \phi \, dx = -\int_{S_h(t)} I(\nabla_{S_h(t)} X, \nabla_{S_h(t)} \phi) \, dx \qquad t > 0 \tag{10}$$

$$M_h(0) = M_{h_0} \tag{11}$$

for all $\phi \in V_h^3$ with $\phi = 0$ on ∂S_h. We choose the subspace consisting of linear polynomials on each triangle.

$$V_h = \{\phi \in C^0(M_h) \text{ s.t. } |\phi|_T \text{ is a linear polynomial on each triangle } T\}.$$

Choosing time step τ and letting $X^n = X(n\tau)$ and $S_h^n = S_h(n\tau)$, we discretize semi-implicitly in time [17] by

$$\int_{S_h^n} \frac{X^{n+1} - X^n}{\tau} \phi \, dx = -\int_{S_h^n} I(\nabla_{S_h^n} X^{n+1}, \nabla_{S_h^n} \phi) \, dx \qquad n \in [0, 1, \ldots) \tag{12}$$

$$M_h^0 = M_{h_0}, \tag{13}$$

for all $\phi \in V_h^3$ such that $\phi = 0$ on ∂S_h^n. We thus observe that the surface S_h^{n+1} at time step $n+1$ is parameterized over the surface at the previous time step S_h^n. For each time step n, we choose the nodal basis $\{\Phi_i^n\}_{i=1}^m$ for V_h, where m is the number of vertices. That is, for each vertex $\bar{X}_i^n$ we associate a piecewise linear basis function Φ_i^n such that

$$\Phi_j^n(\bar{X}_i^n) = \delta_{ij}, \quad i, j = 1, \ldots, m$$

The coordinate map can now be written in terms of the basis as $X^{n+1} = \sum_{i=1}^m \bar{X}_i^{n+1} \Phi_i^n$ and $X^n = \sum_{i=1}^m \bar{X}_i^n \Phi_i^n$. Using these expressions, equation (12) can be written as a linear system for the new vertex coordinates $\bar{X}^{n+1}$.

$$(M + \tau L)\bar{X}^{n+1} = M\bar{X}^n \qquad n \in [0, 1, \ldots) \tag{14}$$

where $M_{ij} = \int_{S_h^n} \Phi_i^n \Phi_j^n \, dx$ and $L_{ij} = \int_{S_h^n} I(\nabla_{S_h^n} \Phi_i^n, \nabla_{S_h^n} \Phi_j^n) \, dx$. At each time step, this linear system must be solved for each of the three vertex coordinates. Note that the stiffness matrix $(M + \tau L)$ is identical for each of the three linear systems, only the right hand side differs.

By the choice of the basis, we observe that $(M + \tau L)$ is very sparse, with non-zeroes mainly concentrated around the diagonal. To numerically solve the linear systems, iterative methods like the conjugate gradient method (CG) or SOR can be used. At each time step n, $\bar{X}^n$ can be used for initialization, since we expect $\bar{X}^{n+1}$ to be close to $\bar{X}^n$.

4 Surface Compression

The new compression method can now be formulated. A triangular mesh is encoded by deleting all vertices that are reconstructed in acceptable quality by the PDE. Only the remaining vertices are stored on the computer. In the decoding step, the unknown vertices are reconstructed via PDE based interpolation.

4.1 Encoding

To achieve a compression rate of p percent, we aim to store only p percent of the vertices. The vertices should be selected such that the reconstruction error when using them for interpolation points is minimized. Intuitively, these points should be located at areas of high curvature. To efficiently find exact or suboptimal solutions of this combinatorial optimization problem is of interest in its own. Since the goal of this paper is not efficiency, but to demonstrate the potential of the method, we present a stochastic approach. The idea is to progressively reduce the number of vertices. This is achieved by interpolating in small randomly selected subsets of the vertices, and permanently removing the vertices with least reconstruction error:

Encoding Algorithm:
Set $V = \{\text{all vertices in the mesh}\}$.
Repeat until desired sparsification is reached:

1. Randomly remove set R of vertices (e.g. $|R| = 5\%$ of $|V|$).
2. Interpolate in $V \backslash R$ with geometric diffusion.
3. Permanently remove the set P, where reconstruction error is smallest (e.g. $|P| = 5\%$ of $|R|$)

$$V \leftarrow V \backslash P$$

In the interpolation step (step 2), the vertices in the known surface are used for initialization. Finally, a coarse triangulation topologically equivalent to the original mesh should be defined for the sparse set of vertices. This triangulation should be stored in addition to the set of vertices.

It is hard to assess the exact compression rate of our method at the current stage. We assume that if p percent of the vertices are to be stored, the coarse triangulation requires p percent as much storage as the original triangulation. Therefore, the compression rate is also p percent. Since the coarse triangulation can potentially be stored with much less space, the actual compression should be even higher.

4.2 Decoding

The surface is decoded by interpolating from the stored set of vertices V. Note that we do not aim to reconstruct the exact locations of the original vertex positions. Our aim is rather to reconstruct a surface which as close as possible resembles the original surface. Within that surface, we do not care how the vertices are distributed relative to the original surface.

Since only a small amount of vertices are known, the interpolation method requires an initialization of all the unknown vertices. Furthermore an initial triangulation must be defined on the set of all known and initial vertices. Note that such an initialization, also determines an initialization of the metric coefficients of the discrete Laplace-Beltrami operator.

The initialization can be computed in many ways. Starting with the coarse mesh, a possibility is to iteratively refine and solve the PDE, introducing more vertices in each iteration. At each step, the mesh should be topologically equivalent to the stored mesh.

In our experiments, we have simply used the known mesh for initialization. This will not have any influence on the final result, since under Dirichlet boundary conditions, the steady state of the geometric diffusion equation is unique.

5 Surface and Image Approximation

It is well known that the diffusion equation results in a smooth functions everywhere, except at the boundary where we in general cannot say anything about the smoothness. In fact, as the area of the boundary points goes to zero, a logarithmic singularity will arise at each boundary point. For the interpolation problem, this can be observed as a sharp cusp at each interpolation point, although in the discrete setting the singularities are not fully developed. The phenomena happens both for surfaces and images, but is much more visible for surfaces.

We now present a method to counteract this problem, which is based on approximation instead of interpolation. Afterwards, a further improvement is presented which is also able to prevent shrinking effects, and yields a sharper reconstruction at edges.

The idea is to iteratively exchange the role of interpolation domain and boundary domain. Letting S_1 denote the set of unknown vertices, and S_2 the set of known vertices, the algorithm can be sketched as follows.

Modification I:
for n=1,...,N:
1. Interpolate S_1^{n+1} with S_2^n fixed
2. Interpolate S_2^{n+1} with S_1^{n+1} fixed

where S_i^0 is the input and S_i^N is the output for $i = 1,2$.

This process may be interpreted as Hopscotch iteration for the geometric diffusion equation in the whole domain. Hopscotch iteration was developed in the 1960's as a numerical method for solving time dependent PDEs [26,27]. The idea is to decompose the spatial domain into two or more subdomains consisting of scattered regions. The PDE is solved numerically by alternatingly taking one time step in each subdomain, with function values in the inactive subdomain as boundary conditions. Attempts to analyze the numerical error has been made in the above mentioned references. For a given compression rate, S_1 and S_2 each contain a certain percentage of vertices. Therefore, this method is independent on the resolution of the mesh.

Note that this method achieves smoothness at the expense of accuracy, as the surface will be progressively diffused in each time step. However, the smoothness will be significantly improved after only one iteration.

A further improvement can be made. For decompression, we observe that the interpolation points are located at locations of sharp edges and corners, where the curvature is large. We therefore propose the same iterative scheme, but instead project the displacements of the interpolation points back in their opposite direction. This can be seen as inverse diffusion applied to the interpolation points. The motivation is to imitate the behaviour of edge enhancing nonlinear PDEs, which perform inverse diffusion at locations of sharp corners. In order to maintain a smooth result, backward diffusion is only applied every second iteration. The algorithm is sketched below.

Modification II:
for n=1,2,...,N:
1. Interpolate S_1^{n+1} with S_2^n fixed.
2. Interpolate S_2^{n+1} with S_1^{n+1} fixed.
3. if n is odd: $S_2^{n+1} \leftarrow S_2^n - 2S_2^{n+1}$

We experienced that usually the best results were obtained with one iteration of modification I, or two iterations of modification II. All experiments in this paper are made with these number of iterations.

We have also applied this method for image compression with linear diffusion. In this case S_1 and S_2 can be replaced by Ω_1 and Ω_2, where Ω_1 is the domain of unknown pixels, and Ω_2 is the domain of known pixels. Analyzing the L1 error, we are able to obtain equally good reconstruction results based on linear diffusion as the more expensive nonlinear anisotropic diffusion.

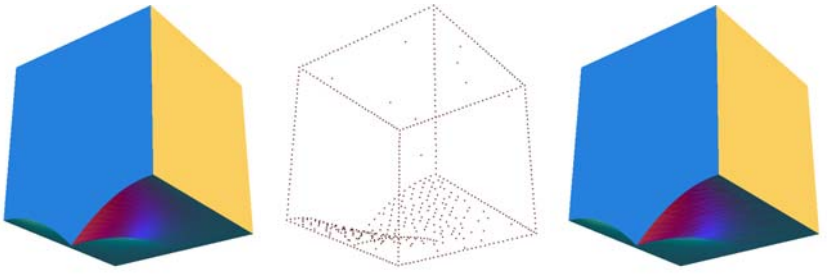
6 Numerical Experiments

This section presents numerical results. The mean L2 error measures for all surface experiments, computed by the metro tool [28], are shown in Table 1. An illustrative example is shown in Figure 1, where 10 percent of the interpolation vertices have been selected. The coordinates of these vertices are depicted as dark dots in Figure 1(b). This demonstrates that the encoding step tends to select vertices at locations of high curvature, such as sharp edges. For this particular example, the compression is not lossy, as the exact surface can be reconstructed.

The next example, Figure 4, aims to show the differences between the interpolation method and the different approximation modifications. For this low resolution mesh, the sharp cusps at each interpolation point are particularly visible, Figure 4 (b). As seen in Figure 4(c)(d), the two modifications significantly reduces the problem. In addition, modification II yields a sharper reconstruction with less shrinking effects. Although it may be hard to see this difference when the surfaces are placed next to each other, the effect is clear from the error measures in Table 1, Moai. As expected, modification I

Table 1. Mean L2 errors (metro) for surfaces

	Compression rate	Geometric diffusion	Modification I	Modification II
Moai	2 percent	-	-	-
	5 percent	0.047	0.050	0.024
	10 percent	0.021	0.024	0.009
Armadillo	2 percent	0.218	0.226	0.171
	5 percent	0.128	0.138	0.082
	10 percent	0.065	0.079	0.049
Max-Planck	2 percent	0.107	0.112	0.091
	5 percent	0.074	0.082	0.056
	10 percent	0.045	0.059	0.033



(a) Ground truth (b) interpolation points (c) Geometric diffusion

Fig. 1. Reconstruction from 10% of vertices

yields slightly higher errors than geometric diffusion, while modification II is significantly better than both geometric diffusion and modification I.

Some more realistic examples are depicted next. In Figure 5, we compare interpolation with geometric diffusion, modification I, and modification II. For this example, 90 percent of the vertex coordinates have been deleted in the encoding step. We observe that the overall shape can be reconstructed quite well with geometric diffusion, Figure 5(b). A disadvantage is the sharp cusps at each interpolation point. The result of one iteration of modification I is shown in Figure 5(c). The surface is smoother, but has the drawback of smooth corners and edges, and slight shrinking effects. Figure 5(d) is the result of two iterations of modification II, which we observe yields both a sharper results at edges, while removing singularities at the boundary points. See also Table 1, Armadillo.

Another example is shown in Figure 6, where 95 percent of the vertex coordinates have been removed. This example demonstrates that quite amazing results can be obtained with geometric diffusion. The improved result with modification II is also shown. At such high compression rate some small scale detail, such as wrinkles, may become less prominent.

We also demonstrate the approximation method presented in Section 5 for image compression. In Figure 2 and 3, the reconstruction results of linear diffusion, anisotropic diffusion and modification II are shown. In Table 2, we compare the L1 reconstruction

Fig. 2. Input images. Left: Trui, Right: Peppers. Resolution: 256×256.

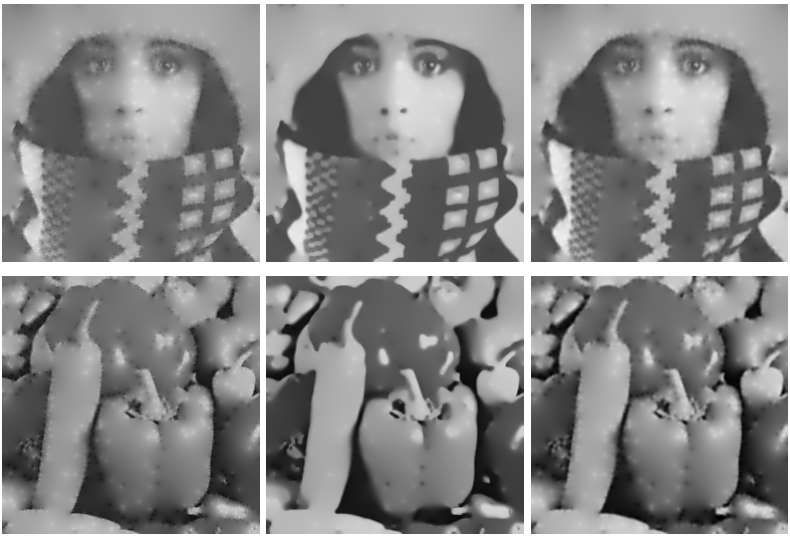
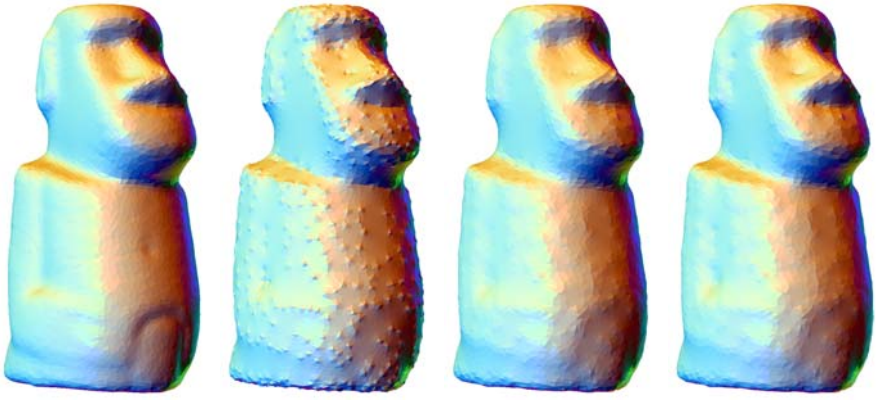


Fig. 3. Reconstructions from 2.5 % of pixels. Top: Trui. Bottom: Peppers. From left to right: linear diffusion, anisotropic diffusion, modification II.



(a) Ground truth (b) Geometric diffusion (c) Modification I (d) Modification II

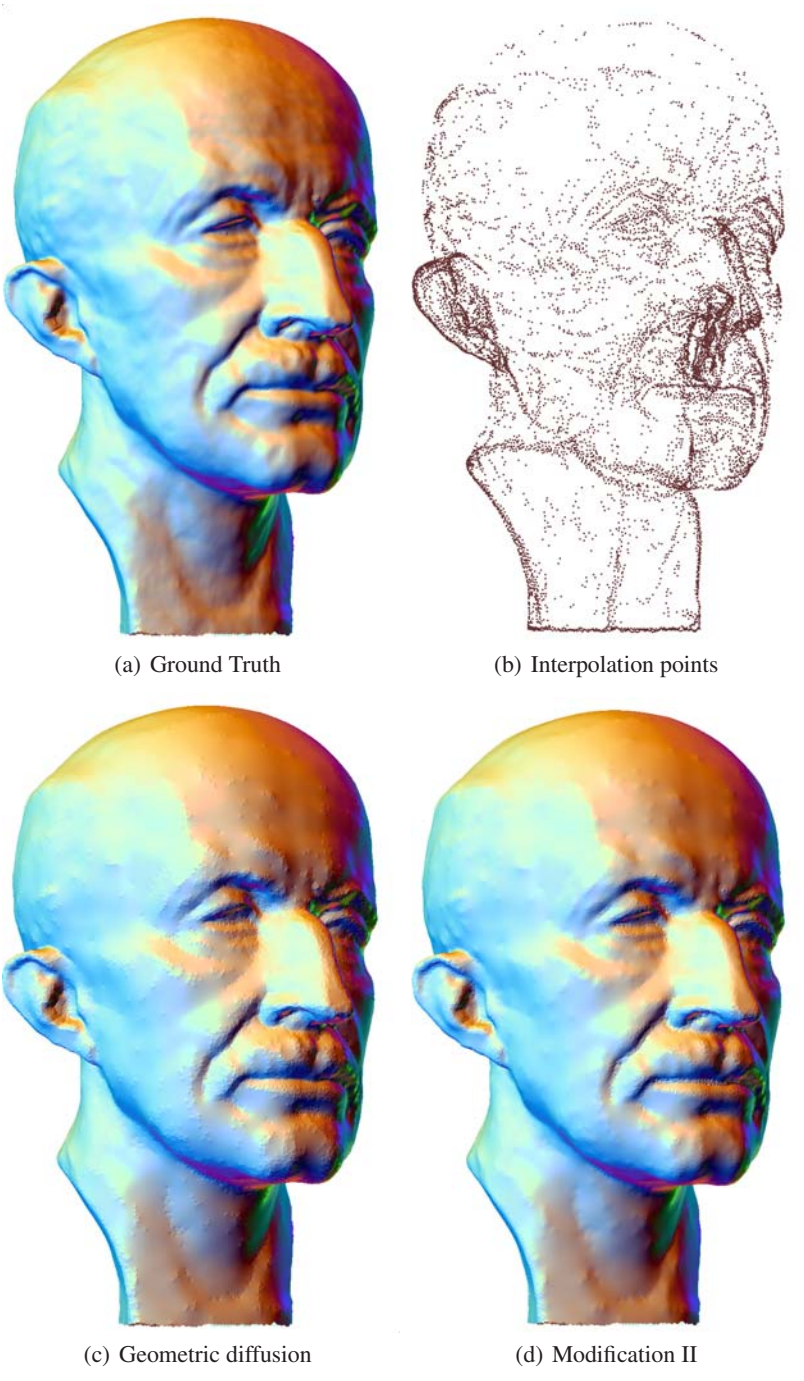
Fig. 4. Reconstruction from 10% of vertices, Moai experiment. Total number of vertices: 10002, total number of triangles: 20000. (a) Ground Truth, (b) interpolation by geometric diffusion, (c) modification I, (d) modification II.

error of the different PDEs. We observe that modification II yields equally good results as the more sophisticated anisotropic diffusion, although it seems to have a slightly lesser ability of preserving edges. Anisotropic diffusion was shown to outperform the well known JPEG standard in [4]. Since modification II is based on linear diffusion, it is several times faster than anisotropic diffusion.

(a) Ground truth (b) Geometric diffusion

(c) Modification I (d) Modification II

Fig. 5. Reconstruction from 10% of vertices, Armadillo man experiment. Total number of vertices: 165954, total number of triangles: 331904. (a) Ground Truth, (b) interpolation by geometric diffusion, (c) modification I, (d) modification II.



(a) Ground Truth (b) Interpolation points

(c) Geometric diffusion (d) Modification II

Fig. 6. Reconstruction from 5% of vertices, Max Planck experiment. Total number of vertices: 199169, total number of triangles: 398043. (a) Ground Truth, (b) selected interpolation points, (c) interpolation geometric diffusion, (d) modification II.

Table 2. Average absolute errors (L1 errors) for images

	Linear diffusion	Anisotropic diffusion	Modification II
Trui	14.614	7.697	8.037
Peppers	12.967	10.591	9.471

7 Conclusions

We have presented new PDE based methods for interpolation, approximation and compression of triangulated surfaces. Experiments show that amazingly good results can be obtained with the simple geometric diffusion equation acting on the coordinate map from unoriented points (i.e. without any normal vectors) when the interpolation points are selected properly.

This work may only be considered a proof of concept, as further developments are necessary for a practical method. In the future, we will study anisotropic geometric diffusion equations [19] and develop a faster algorithm for selecting the best interpolation points in the encoding step. Comparisons with other methods will also be made.

Acknowledgements

Thanks to Martin Welk, Alexander Belyaev and Michael Breuss for interesting discussions. We also thank Alexander Belyaev for providing useful software and database of meshes.

References

1. Khodakovsky, A., Schroeder, P., Sweldens, W.: Progressive geometry compression. In: Proceedings of Siggraph, pp. 95–102 (2000)
2. DeVore, R.A., Jawerth, B., Lucier, B.J.: Surface compression. Computer Aided Geometric Design 9, 219–239 (1992)
3. Davis, G.M., Nosratinia, A.: Wavelet-based image coding: An overview. In: Datta, B.N. (ed.) Applied and Computational Control, Signals, and Circuits, vol. 1, pp. 205–269. Birkhäuser, Boston (1999)
4. Galić, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A., Seidel, H.P.: Image compression with anisotropic diffusion. Journal of Mathematical Imaging and Vision 31, 255–269 (2008)
5. Masnou, S., Morel, J.M.: Level lines based disocclusion. In: 5th IEEE International Conference on Image Processing, pp. 259–263 (1998)
6. Bertalmio, M., Sapiro, G., Ballester, C.: Image inpainting. In: Proc. SIGGRAPH 2000, pp. 417–424 (2000)
7. Chan, T.F., Shen, J.: Non-texture inpainting by curvature-driven diffusions (CDD). Journal of Visual Communcation and Image Representation 12, 436–449 (2001)
8. Chan, T.F., Ha, S., Kang, S.J.: Euler's elastica and curvature based inpaintings. SIAM Journal of Applied Mathematics 63, 564–592 (2002)
9. Verdera, J., Bertalmio, M., Sapiro, G.: Inpainting surface holes. In: Int. Conference on Image Processing, pp. 903–906 (2003)

10. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: SGP 2006: Proceedings of the fourth Eurographics symposium on Geometry processing, Aire-la-Ville, Switzerland, pp. 61–70. Eurographics Association (2006)
11. Zhao, H.K., Osher, S., Fedkiw, R.: Fast surface reconstruction using the level set method. In: VLSM 2001: Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM 2001), Washington, DC, USA, pp. 194–201. IEEE Computer Society, Los Alamitos (2001)
12. Kobbelt, L.: Discrete fairing. In: Proceedings of the Seventh IMA Conference on the Mathematics of Surfaces, pp. 101–131 (1997)
13. Schneider, R., Kobbelt, L.: Discrete fairing of curves and surfaces based on linear curvature distribution. In: Curve and Surface Design, pp. 371–380. University Press, Saint-Malo (2000)
14. Weickert, J.: Theoretical foundations of anisotropic diffusion in image processing. Computing (suppl. 11), 221–236 (1996)
15. Belhachmi, Z., Bucur, D., Burgeth, B., Weickert, J.: How to choose interpolation data in images. Technical Report No. 205, Department of Mathematics, Saarland University, Saarbrücken, Germany (2008)
16. Desbrun, M., Meyer, M., Schröder, P., Barr, A.H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: SIGGRAPH 1999: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pp. 317–324. ACM Press/Addison-Wesley Publishing Co., New York (1999)
17. Dziuk, G.: An algorithm for evolutionary surfaces. Numerische Mathematik 58, 603–611 (1991)
18. Dzuik, G.: Finite elements for the Beltrami operator on arbitrary surfaces. Partial Differential Equations and Calculus of Variations 1357, 142–155 (1988)
19. Clarenz, U., Diewald, U., Rumpf, M.: Anisotropic geometric diffusion in surface processing. In: VIS 2000: Proceedings of the conference on Visualization 2000, pp. 397–405. IEEE Computer Society Press, Los Alamitos (2000)
20. Taubin, G.: A signal processing approach to fair surface design. In: SIGGRAPH 1995: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pp. 351–358. ACM, New York (1995)
21. Sorkine, O., Cohen-Or, D., Toledo, S.: High-pass quantization for mesh encoding. In: SGP 2003: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing, Aire-la-Ville, Switzerland, Switzerland, pp. 42–51. Eurographics Association (2003)
22. Carmo, M.P.D.: Riemannian geometry. Birkhäuser, Basel (1993)
23. Sapiro, G.: Geometric partial differential equations and image analysis. Cambridge University Press, Cambridge (2001)
24. Tasdizen, T., Whitaker, R., Burchard, P., Osher, S.: Geometric surface processing via normal maps. ACM Trans. Graph. 22(4), 1012–1033 (2003)
25. Bobenko, A.I., Schröder, P.: Discrete willmore flow. In: SIGGRAPH 2005: ACM SIGGRAPH 2005 Courses, p. 5. ACM, New York (2005)
26. Gordon, P.: Nonsymmetric difference schemes. Journal of the Society of Industrial and Applied Mathematics 13(3), 667–673 (1965)
27. Gourlay, A.: Hopscotch: a fast second-order partial differential equation solver. IMA Journal of Applied Mathematics 6(4), 375–390 (1969)
28. Cignoni, P., Rocchini, C., Scopigno, R.: Metro: measuring error on simplified surfaces. In: Computer Graphics Forum, vol. 17(2), pp. 167–174. Blackwell Publishers, Malden (1998)

Construction of Rational Curves with Rational Rotation-Minimizing Frames via Möbius Transformations

Michael Bartoň[1], Bert Jüttler[2], and Wenping Wang[3]

[1] Technion - Israel Institute of Technology, Haifa, Israel
[2] Institute of Applied Geometry, Johannes Kepler University, Linz, Austria
[3] Dept. of Computer Science, University of Hongkong, China

Abstract. We show that Möbius transformations preserve the rotation-minimizing frames which are associated with space curves. In addition, these transformations are known to preserve the class of rational Pythagorean-hodograph curves and rational frames. Based on these observations we derive an algorithm for G^1 Hermite interpolation by rational Pythagorean-hodograph curves with rational rotation-minimizing frames.

Keywords: rational rotation-minimizing frame, Möbius transformations, Pythagorean-hodograph curve.

1 Introduction

The rotation–minimizing frames which are associated with a space curve are useful for generating sweep surfaces by moving a profile curve along a trajectory. The first publication discussing this frame was probably [4]. Since then, the computation of this frame and of the resulting sweep surfaces has been discussed in a substantial number of publications, see [23] and the references cited therein. In particular, the construction of rational parameterizations of these sweep surfaces is of interest, due to the importance of NURBS representations for Computer-Aided Design.

The class of spatial Pythagorean–hodograph (PH) curves, which was introduced in [10], seems to be particularly useful for sweep surface modeling. See [8] for an introduction to this topic with many related references. These curves are equipped with rational frames [14], which are called the Euler-Rodrigues frames [5, 6]. In addition, certain PH curves support the exact computation of rotation–minimizing frames via integration of rational functions [7].

PH space curves with rational rotation-minimizing frames have also been analyzed in the literature. While such frames do not exist on PH cubics [12], an example of a spatial PH curve of degree 7 with a rational rotation-minimizing frame has been reported in [5]. In that paper, the authors analyzed the conditions under which the Euler-Rodrigues frame of a PH curve is also a rotation–minimizing frame, and they showed that this requires at least polynomial degree 7 for space curves. In a recent manuscript, [9], the authors use the Hopf map representation for PH space curves to demonstrate the existence of rational rotation-minimizing frames on spatial PH quintics.

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 15–25, 2010.

The surfaces generated by rotation-minimizing frames have been called profile surfaces in [18,19]. Based on the kinematical interpretation of rotation-minimizing frames as plane-rolling motions, a construction for rational profile surfaces has been presented in [19]. This construction gives rational space curves with rational rotation-minimizing frames, but the degree of the frames – and hence of the resulting profile surfaces – is relatively high for non-planar curves.

Möbius transformations map polynomial or rational PH curves into rational PH curves [20], where a rational PH curve is defined as a curve with rational parametric speed. In addition to this result, in the present paper we show that Möbius transformations preserve rotation-minimizing frames, and in particular rational ones. This observation greatly enhances the flexibility of existing constructions for curves with rational rotation-minimizing frames [5,9], since the group of Möbius transformations can be used to generate new ones.

We demonstrate this observation by formulating an algorithm for G^1 Hermite interpolation by rational PH curves with rational rotation-minimizing frames. It is based on Möbius transformations which are applied to planar PH cubics. In the case of the latter curves, the Frenet frames are trivially rotation-minimizing and also rational.

2 Rotation-Minimizing Frames and Möbius Transformations

We recall the notion of the rotation-minimizing frame which can be associated with any space curve and summarize the properties of Möbius transformations. We then analyze the behaviour of rotation–minimizing frames, of Pythagorean-hodograph curves, and of curves with rational rotation-minimizing frames under Möbius transformations.

2.1 Rotation-Minimizing Frames

We consider a C^∞ parametric curve segment $\mathbf{x} : u \mapsto \mathbf{x}(u)$ with the parameter domain $I = [0,1]$. We assume that $\mathbf{x}$ is regular for all $u \in I$, i.e., $\mathbf{x}'(u) \neq \mathbf{0}$, where the prime $'$ indicates the differentiation with respect to u.

We consider a vector field $\mathbf{r} : u \mapsto \mathbf{r}(u)$ along the curve $\mathbf{x}$. Such a vector field is said to be a *rotation–minimizing* with respect to $\mathbf{x}$ if there exists a function λ such that $\mathbf{r}$ satisfies the equations

$$\left.\begin{aligned} \mathbf{r}'(u) - \lambda(u)\mathbf{x}'(u) &= 0 \\ \mathbf{x}'(u) \cdot \mathbf{r}(u) &= 0 \end{aligned}\right\} \tag{1}$$

for all $u \in I$. The first equation implies $\lambda = (\mathbf{r}' \cdot \mathbf{x}')/(\mathbf{x}' \cdot \mathbf{x}')$. The second equation expresses the fact that the vector $\mathbf{r}(u)$ is contained in the normal plane of the curve at $\mathbf{x}(u)$. Due to

$$(\mathbf{r}(u) \cdot \mathbf{r}(u))' = 2\mathbf{r}'(u) \cdot \mathbf{r}(u) = 2\lambda(u)\mathbf{x}'(u) \cdot \mathbf{r}(u) = 0, \tag{2}$$

the length of the rotation-minimizing vectors $\mathbf{r}(u)$ is constant.

A *rotation–minimizing frame* (RMF) of the given curve $\mathbf{x}$ is defined as a triplet of vectors

$$u \mapsto (\mathbf{t}(u), \mathbf{r}(u), \mathbf{s}(u)) \tag{3}$$

with a unit vector $\mathbf{r}(u)$, where $\mathbf{t} = \mathbf{x}'/|\mathbf{x}'|$ is the unit tangent vector of the given curve and $\mathbf{s}(u) = \mathbf{t}(u) \times \mathbf{r}(u)$. Any vector field $\rho\mathbf{r}(u) + \sigma\mathbf{s}(u)$ with constant coefficients ρ, σ is obviously again rotation–minimizing.

Let

$$F_{\mathbf{x}} = \{\mathbf{w}, \mathbf{w} \cdot \mathbf{x}'(0) = 0, \|\mathbf{w}\| = 1\} \tag{4}$$

be the unit circle in the normal plane of the curve at $\mathbf{x}(0)$, which is also the set of feasible initial values for rotation-minimizing frames defined by the equations (1) at $u = 0$.

Finally we define the *rotation–minimizing operator*

$$\mathscr{R}_{\mathbf{x}} : \quad F_{\mathbf{x}} \to \mathscr{C}(I, \mathbb{S}^2), \tag{5}$$

where $\mathbb{S}^2$ is the unit sphere in $\mathbb{R}^3$, as the mapping which assigns to any initial vector $\mathbf{r}_0 \in F_{\mathbf{x}}$ the solution of the equations (1) with initial value $\mathbf{r}(0) = \mathbf{r}_0$. Consequently, $\mathscr{R}_{\mathbf{x}}(u)$ is a mapping from $F_{\mathbf{x}}$ into $\mathbb{S}^2$ for any fixed value of the parameter $u \in I$.

2.2 Möbius Transformations

We consider the three–dimensional space which is extended by adding a single point ∞ at infinity. For any sphere with radius $r > 0$ and center $\mathbf{c}$, the mapping

$$\rho : \mathbf{y} \mapsto \mathbf{c} + \frac{r^2}{\|\mathbf{y} - \mathbf{c}\|^2}(\mathbf{y} - \mathbf{c}), \quad \mathbf{y} \notin \{\mathbf{c}, \infty\}, \tag{6}$$

which additionally maps $\mathbf{c}$ to ∞ and vice versa, is said to be the *inversion* with respect to the sphere. It satisfies $\rho^2 = \mathrm{id}$. The reflection at a plane is also considered as an inversion, which keeps ∞ fixed. A general *Möbius transformation* μ is defined as a composition of an arbitrary number of inversions with respect to spheres or planes. In particular, translations and rotations are special Möbius transformations, since they can be obtained by composing two reflections at planes.

The set of Möbius transformations forms a group with respect to composition, which contains the Euclidean group as a subgroup. Möbius transformations are known to be the most general conformal (i.e., angle-preserving) transformations of the three–dimensional space. The image of any plane or sphere under a Möbius transformation is again a plane or a sphere.

While the Euclidean group provides six degrees of freedom, the set of Möbius transformations depends on 10 independent parameters. These transformations can be represented by matrices of the group $O(4,1)$, which is a Lie group of dimension 10. Any Möbius transformation is also a birational mapping. See [1, 2, 15] for additional information on this interesting class of transformations.

The *differential*

$$\mathrm{d}\mu_{\mathbf{y}} : \mathbb{R}^3 \to \mathbb{R}^3 \tag{7}$$

of a Möbius transformation at a point $\mathbf{y}$ maps the tangent space of the three dimensional-space at any point $\mathbf{y}$ into the tangent space at $\mu(\mathbf{y})$. More precisely, for any vector $\mathbf{v}$ and any differentiable function $f : \mathbb{R}^3 \to \mathbb{R}$, the directional derivative of f with respect to the direction $\mathrm{d}\mu_{\mathbf{y}}(\mathbf{v})$ at the point $\mu(\mathbf{y})$ is equal to the directional derivative of $f \circ \mu$ with respect to the direction $\mathbf{v}$ at the point $\mathbf{y}$, see [11].

In addition we define the *normalized differential*

$$\mathrm{d}^\star\mu_{\mathbf{y}}:\ \mathbb{R}^3 \to \mathbb{R}^3 : \mathbf{v} \mapsto \frac{\|\mathbf{v}\|}{\|\mathrm{d}\mu_{\mathbf{y}}(\mathbf{v})\|}\mathrm{d}\mu_{\mathbf{y}}(\mathbf{v}) \tag{8}$$

as the version of this mapping which preserves the length of all vectors $\mathbf{v}$.

As a special case we consider the inversion ϕ with respect to the unit sphere,

$$\phi:\ \mathbf{y} \mapsto \frac{1}{\|\mathbf{y}\|^2}\mathbf{y}, \quad \mathbf{y} \notin \{\mathbf{0}, \infty\}. \tag{9}$$

The differential $\mathrm{d}\phi_{\mathbf{y}}$ of this inversion is

$$\mathrm{d}\phi_{\mathbf{y}}:\ \mathbb{R}^3 \to \mathbb{R}^3:\ \mathbf{v} \mapsto \frac{\|\mathbf{y}\|^2\mathbf{v} - 2(\mathbf{v}\cdot\mathbf{y})\mathbf{y}}{\|\mathbf{y}\|^4}. \tag{10}$$

A short computation confirms that the length of the image of a vector equals

$$\|\mathrm{d}\phi_{\mathbf{y}}(\mathbf{v})\| = \frac{\|\mathbf{v}\|}{\|\mathbf{y}\|^2}. \tag{11}$$

We use this result to find the *normalized differential* which preserves the length of the vectors,

$$\mathrm{d}^\star\phi_{\mathbf{y}}(\mathbf{v}):\ \mathbb{R}^3 \to \mathbb{R}^3 : \mathbf{v} \mapsto \|\mathbf{y}\|^2\,\mathrm{d}\phi_{\mathbf{y}}(\mathbf{v}) = \mathbf{v} - \frac{2(\mathbf{v}\cdot\mathbf{y})}{\|\mathbf{y}\|^2}\mathbf{y}. \tag{12}$$

In the case of a general Möbius transformation, which is obtained by composing several inversions, the differential and the normalized differential are both linear transformations whose coefficients are quadratic rational functions of the coordinates of the point $\mathbf{y}$.

2.3 Invariance of Rotation-Minimizing Frames

Möbius transformations commute with the computation of rotation minimizing frames. More precisely, one has the following result.

Theorem 1. *Consider a Möbius transformation μ which maps the curve $\mathbf{x}$ into another regular curve segment $\mu \circ \mathbf{x}$. Then*

(i) the evaluation of the rotation-minimizing vector field with respect to $\mathbf{x}$ at the parameter value u for an initial vector $\mathbf{r}_0$ and the application of the normalized differential of μ to the resulting vector

gives the same result as

(ii) the evaluation of the rotation-minimizing vector field with respect to $\mu \circ \mathbf{x}$ at this parameter value u, where the initial vector is the image of $\mathbf{r}_0$ under the normalized differential of μ at $\mathbf{x}(0)$,

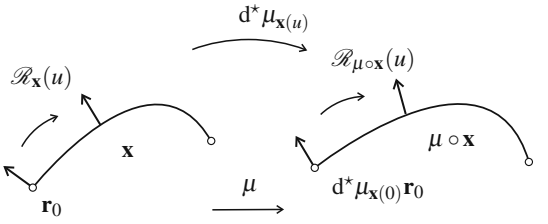


Fig. 1. Invariance of RMF under Möbius transformations, see Theorem 1

cf. Fig. 1. More precisely,

$$\mathscr{R}_{\mu\circ\mathbf{x}}(u)\circ\mathrm{d}^\star\mu_{\mathbf{x}(0)}=\mathrm{d}^\star\mu_{\mathbf{x}(u)}\circ\mathscr{R}_{\mathbf{x}}(u)\quad \textit{for all } u\in I, \tag{13}$$

where $\mathscr{R}_{\mu\circ\mathbf{x}}$ and $\mathscr{R}_{\mathbf{x}}$ are the rotation-minimizing operators of the transformed and the original curve and $\mathrm{d}^\star\mu_{\mathbf{x}(u)}$ is the normalized differential of the Möbius transformation at $\mathbf{x}(u)$, respectively.

Proof. It suffices to prove this result for the inversion ϕ with respect to the unit sphere, since any Möbius transformation is a composition of inversions and the result is obviously true for uniform scalings and reflections at planes.

We consider a rotation-minimizing vector $\mathbf{v}$ of the given curve $\mathbf{x}$. Consequently, it satisfies the two conditions

$$\mathbf{v}\cdot\mathbf{x}'=0\quad\text{and}\quad\mathbf{v}'=\frac{\mathbf{v}'\cdot\mathbf{x}'}{\mathbf{x}'\cdot\mathbf{x}'}\mathbf{x}' \tag{14}$$

see (1). Using a direct computation one can show that

$$\begin{aligned}\mathrm{d}^\star\phi_{\mathbf{x}(u)}(\mathbf{v})\cdot(\phi\circ\mathbf{x})'&=\left(\mathbf{v}-\frac{2(\mathbf{v}\cdot\mathbf{x})}{\|\mathbf{x}\|^2}\mathbf{x}\right)\cdot\left(\frac{\|\mathbf{x}\|^2\mathbf{x}'-2\langle\mathbf{x}',\mathbf{x}\rangle\mathbf{x}}{\|\mathbf{x}\|^4}\right)=\\&=\frac{1}{\|\mathbf{x}\|^4}\left(\|\mathbf{x}\|^2(\mathbf{v}\cdot\mathbf{x}')+(-2-2+4)\frac{\|\mathbf{x}\|^2(\mathbf{x}'\cdot\mathbf{x})(\mathbf{v}\cdot\mathbf{x})}{\|\mathbf{x}\|^2}\right)=0.\end{aligned} \tag{15}$$

In addition we get that

$$\begin{aligned}(\mathrm{d}^\star\phi_{\mathbf{x}(u)}(\mathbf{v}))'&=\left(\mathbf{v}-\frac{2(\mathbf{v}\cdot\mathbf{x})}{\|\mathbf{x}\|^2}\mathbf{x}\right)'=\\&=\mathbf{v}'-\left(\frac{2(\mathbf{v}'\cdot\mathbf{x})}{\|\mathbf{x}\|^2}+\frac{2(\mathbf{v}\cdot\mathbf{x}')}{\|\mathbf{x}\|^2}+\frac{4(\mathbf{v}\cdot\mathbf{x})(\mathbf{x}\cdot\mathbf{x}')}{\|\mathbf{x}\|^4}\right)\mathbf{x}-\frac{2(\mathbf{v}\cdot\mathbf{x})}{\|\mathbf{x}\|^2}\mathbf{x}'\end{aligned} \tag{16}$$

After replacing $\mathbf{v}'$ using (14) and rearranging one arrives at

$$(\mathrm{d}^\star\phi_{\mathbf{x}(u)}(\mathbf{v}))'=\|\mathbf{x}\|^2\left(\frac{\mathbf{v}'\cdot\mathbf{x}'}{\mathbf{x}'\cdot\mathbf{x}'}-\frac{2(\mathbf{v}\cdot\mathbf{x})}{\|\mathbf{x}\|^2}\right)(\phi\circ\mathbf{x})'. \tag{17}$$

Consequently, if $\mathbf{v}$ is a rotation-minimizing vector field with respect to the original curve $\mathbf{x}$, then the images $\mathrm{d}^\star\phi_{\mathbf{x}(u)}(\mathbf{v})$ form a rotation-minimizing vector field with respect to the image curve $\phi \circ \mathbf{x}$, since the conditions which are analogous to (14) are satisfied. This completes the proof, as the initial conditions at $u = 0$ are linked by the normalized differential $\mathrm{d}^\star\phi_{\mathbf{x}(0)}$ at the first point of the curve. □

Remark 2. Möbius transformations are also known to preserve the lines of curvature of surfaces. This observation can be used to derive a geometric proof of the RMF preservation, since the normal vectors of a surface along a curvature line form a rotation-minimizing vector field.

Remark 3. Another frame of a space curve, which is also preserved by Möbius transformations, can be obtained by assembling a triplet of vectors consisting of (1) the unit tangent vector, (2) the unit normal vector of the osculating sphere at the point of the curve, and (3) the cross product of these two vectors. This frame is generally not a rotation-minimizing frame. See also [3] for a discussion of conformally invariant frames of curves.

2.4 Invariance of PH Curves and Rational RMFs

In this section we consider the special case where $\mathbf{x}$ is a polynomial or rational curve. For any rational curve $\mathbf{x}$ and any rational vector field $\mathbf{r}(u)$, the curve $\mu \circ \mathbf{x}$ is again a rational curve and the vector field $u \mapsto \mathrm{d}^\star\mu_{\mathbf{x}(u)}(\mathbf{r}(u))$ is again a rational vector field. Indeed, Möbius transformations μ are birational mappings and their normalized differentials $\mathrm{d}^\star\mu_{\mathbf{y}}$ are linear transformations whose coefficients depend rationally on the point coordinates of $\mathbf{y}$.

Recall that a curve $\mathbf{x}$ is said to be a polynomial Pythagorean-hodograph (PH) curve if the length $\|\mathbf{x}'\|$ of its first derivative vector is a polynomial function of the parameter u for $u \in I$. Following Ueda [20], we consider the extension of this notion to the rational case: a curve is said to be a rational PH curve if the length $\|\mathbf{x}'\|$ of its first derivative vector is a rational function of the parameter u for $u \in I$.

Corollary 4. *If $\mathbf{x}$ is a polynomial PH curve, then $\mu \circ \mathbf{x}$ is a rational PH curve. Moreover, if $\mathbf{x}$ has a rotation-minimizing frame which is described by rational vector fields $(\mathbf{t}(u), \mathbf{r}(u), \mathbf{s}(u))$, then $\mu \circ \mathbf{x}$ possesses the rational rotation-minimizing frame*

$$(\mathrm{d}^\star\mu_{\mathbf{x}(u)}\mathbf{t}(u), \mathrm{d}^\star\mu_{\mathbf{x}(u)}\mathbf{r}(u), \mathrm{d}^\star\mu_{\mathbf{x}(u)}\mathbf{s}(u)) \tag{18}$$

which is obtained by applying the differential $\mathrm{d}^\star\mu_{\mathbf{x}(u)}$ to these vector fields.

Proof. The first part of the corollary is a direct consequence of the properties of the normalized differential. The second one follows by combining this observation with Theorem 1. □

The first part of this corollary was also noted by Ueda [20, §2]. The degree of a PH curves is doubled by applying a Möbius transformation.

3 Interpolation by Curves with Rational RMF

We describe an algorithm for Hermite interpolation by curves that posses a rational rotation-minimizing frame (i.e. an RMF which can be described by rational vector fields) demonstrate its performance by two examples.

3.1 The Algorithm

Given a curve $\mathbf{x}$ with parameter u and domain $I = [0,1]$, we generate a piecewise rational approximation of its rotation-minimizing frame with k segments, which is globally G^1, as follows (see also Fig. 2).

1. The curve is split into k segments, $u \in [u_{i-1}, u_i]$, where $u_j = j/k$ (Fig. 2a).
2. For each segment we generate the unique sphere which passes through $\mathbf{x}(u_{i-1})$ and $\mathbf{x}(u_i)$ and which also touches the tangents of the curves at these points. Next we choose a Möbius transformation μ_i which transforms this sphere into a plane π_i and apply it to the curve (Fig. 2b,c).
3. For each segment we generate a PH cubic in the plane π_i which interpolates the G^1 Hermite boundary data at $(\mu_i \circ \mathbf{x})(u_{i-1})$ and $(\mu_i \circ \mathbf{x})(u_i)$ and compute its rational rotation-minimizing frame (Fig. 2d).
4. We apply the inverse Möbius transformations μ_i^{-1} to the planar PH cubics and to their rational rotation-minimizing frames (Fig. 2e,f).

The last three steps steps are now explained in more detail.

Step 2. There are two degrees of freedom for choosing the point $\mathbf{f}_i$ on the sphere which is mapped into ∞ by the Möbius transformation. We choose it as the point on the sphere which possesses the maximum distance to $\mathbf{x}(u_{i-1})$ and $\mathbf{x}(u_i)$, in order to obtain a mapping which possesses a relatively small distortion in the region of interest. We define the image plane π_i of the sphere as the unique plane which passes through $\mathbf{x}(u_{i-1})$ and $\mathbf{x}(u_i)$ and has the normal vector $\mathbf{f}_i - \mathbf{z}_i$, where $\mathbf{z}_i$ is the center of the sphere.

Step 3. The Bézier form of PH cubics is characterized by a simple geometric condition on the shape of the control polygon [10]. Using this observation, an algorithm for G^1 Hermite interpolation can be formulated [13,17], which finds the interpolating PH cubic by solving a single quadratic equation. Note that this equation does not always possess real solutions. However, if the data are sampled from a smooth curve with sufficiently small step size (i.e., sufficiently large k), then real solutions exist and to produce a unique Hermite interpolant with approximation order 4. This can be shown by adapting the analysis in the proof of [16, Theorem 10].

Finally we generate the rational rotation-minimizing frame of a planar PH cubic $\mathbf{c}_i$,

$$(\mathbf{c}_i'/\|\mathbf{c}_i'\|, \mathbf{n}, \mathbf{c}_i'/\|\mathbf{c}'\| \times \mathbf{n}), \tag{19}$$

where $\mathbf{n}$ is a unit normal vector of the plane.

Step 4. By applying the inverse Möbius transformation μ_i^{-1} to $\mathbf{c}_i$ we obtain the spherical rational PH curve $\mu_i^{-1} \circ \mathbf{c}_i$ of degree 6. Its rational rotation-minimizing frame consists of the unit tangent vector, the unit normal vector field of the sphere, and of the cross product of these two vectors.

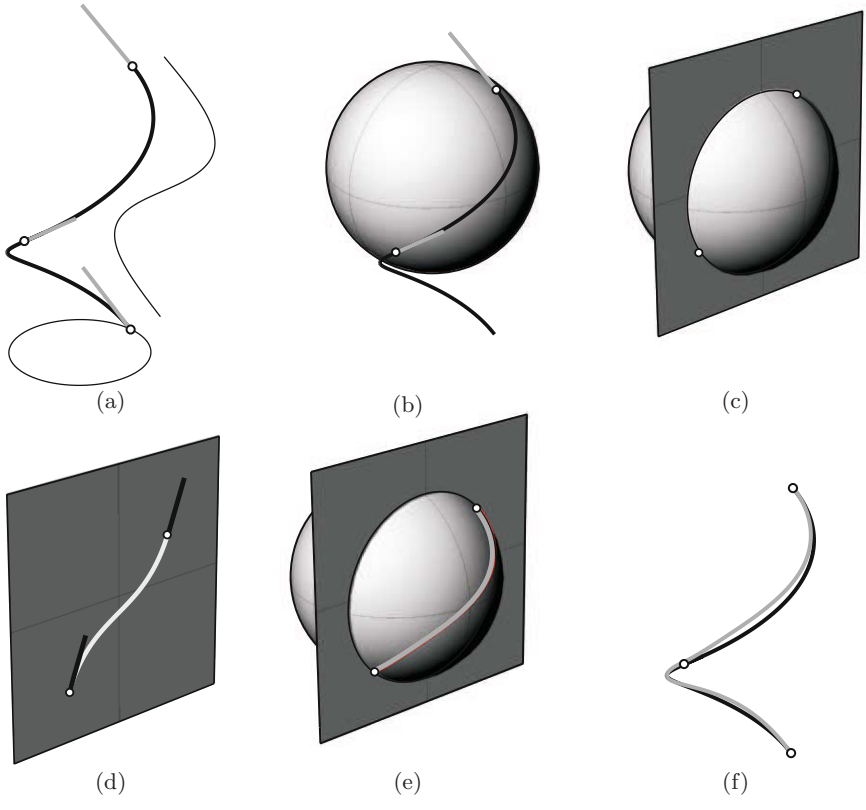


Fig. 2. The algorithm for computation of a rational approximation of the rotation-minimizing frame, applied to a helix, $k = 2$. (a) segmentation, (b,c) defining the Möbius transformation, (d) planar Hermite interpolation, (e) inverse Möbius transformation and (f) helix with approximating curves.

3.2 Examples

Example 5. We apply the algorithm to the curve segment

$$\mathbf{x}(t) = \left(\sin(2t)\cos(t) + \frac{1}{4}\sin(t), \sin(2t)\sin(t) + \frac{1}{4}\cos(t), \frac{1}{2}\sin(4t) \right), \qquad (20)$$

with domain $t \in [\frac{\pi}{4}, 2\pi]$. The approximating rational curves with rational RMF and the developable surface strips generated by a vector of the piecewise rational RMF are shown in Figure 3. The approximation with $k = 8$ segments has still a relatively large distance to the original curve (dashed). If an approximation with $k = 16$ segments is used, then both curves can no longer be visually distinguished.

Example 6. We applied the algorithm to the space PH cubic

$$\mathbf{x}(t) = \left(3t(1-t)^2 + 3t^2(1-t) + t^3, (1-t)^3 + 3t(1-t)^2, 2t^3\right) \qquad (21)$$

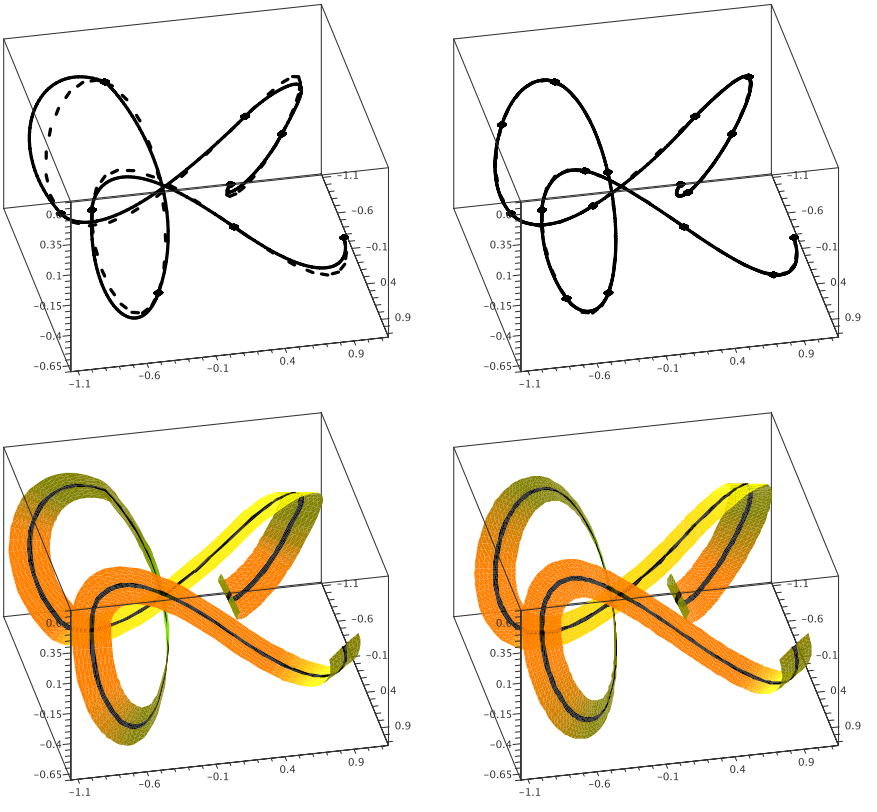


Fig. 3. Top row: Approximation of a space curve (dashed) by rational curves with rational rotation minimizing frames (solid), which were obtained by our algorithm with $k = 8$ (left) and $k = 16$ (right) segments. Bottom row: The strips of developable surfaces which are generated by a vector of the RMF for both approximations.

with domain $t \in [0,1]$. As observed in [7], PH cubics are equipped with rational Frenet frames and the difference angle to the (generally non–rational) rotation-minimizing frame can be computed from

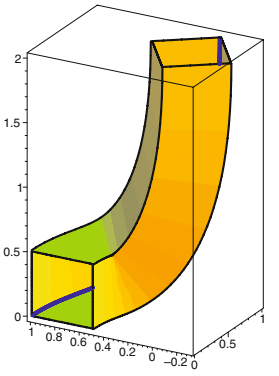
$$\theta(t) = \theta_0 - \int_0^t \tau(u)\,\|\mathbf{x}'(u)\|\,\mathrm{d}u, \tag{22}$$

since both the torsion $\tau(u)$ of $\mathbf{x}$ and the parametric speed $\|\mathbf{x}'(u)\|$ are rational functions of u. Fig. 4 shows the rotation-minimizing frame of (21).

In order to analyze the error, we consider the length of the differences

$$\delta(u) = \|\mathbf{r}_{\text{approx}}(u) - \mathbf{r}(u^*(u))\| \tag{23}$$

of rotation-minimizing vector fields of the approximating curve and of the approximating curve, where each point of the approximating curve is compared with the nearest



k	A_k	$\frac{A_{k/2}}{A_k}$	E_k	$\frac{E_{k/2}}{E_k}$
1	$2.37\ 10^{-1}$	n/a	$2.76\ 10^{-1}$	n/a
2	$3.17\ 10^{-2}$	7.47	$2.67\ 10^{-2}$	10.3
4	$6.86\ 10^{-3}$	4.62	$2.01\ 10^{-3}$	13.3
8	$6.29\ 10^{-4}$	10.91	$1.32\ 10^{-4}$	15.2
16	$1.01\ 10^{-4}$	6.21	$8.35\ 10^{-6}$	15.8
32	$1.30\ 10^{-5}$	7.80	$5.24\ 10^{-7}$	15.9

Fig. 4. Example 6: Sweep surface generated by the non-rational RMF of a space PH cubic (left). Average and end-point error for different numbers of segments (right).

point of the original curve,

$$u^*(u) = \arg \min_{v \in [0,1]} \|\mathbf{x}_{\text{approx}}(u) - \mathbf{x}(v)\|. \tag{24}$$

We use numerical integration to evaluate the average error and the end point error,

$$A_k = \int_0^1 \delta(u) \mathrm{d}u \quad \text{and} \quad E_k = \delta(1) \tag{25}$$

for different numbers of segments k. The results are reported in the table in Fig. 4. The numbers indicate that the average error decreases as h^3, where $h = 2^{-k}$ is the stepsize. The error at the joints – and in particular at the end point of the curve – decreases faster with order 4 (in accordance with the phenomenon which is described in [22,23]).

4 Conclusion

We showed that rotation-minimizing frames of space curves are invariant under Möbius transformations. Since these transformations also preserve rational frames and rational PH curves, it was possible to use this result in order to formulate an algorithm for G^1 Hermite interpolation by rational curves with rational rotation-minimizing frames. As a possible topic for future research, one might try to use the recent results about PH quintics with rational rotation-minimizing frames [9] to achieve higher geometry flexibility, which might lead to a scheme for interpolation with a higher order of smoothness.

Acknowledgment The authors would like to thank Martin Peternell for useful discussions. The first author was supported by grant no. P17387-N12 of the Austrian Science Fund (FWF).

References

1. Ahlfors, L.V.: Möbius transformations in $\mathbb{R}^n$ expressed through 2×2 matrices of complex numbers. Complex Variables 2, 215–224 (1986)
2. Beardon, A.F.: Continued Fractions, Möbius transformations and Clifford Algebras. Bull. London Math. Society 35, 302–308 (2003)
3. Beffa, G.M.: Poisson brackets associated to the conformal geometry of curves. Trans. Amer. Math. Society 357, 2799–2827 (2005)
4. Bishop, R.: There is more than one way to frame a curve. Amer. Math. Monthly 82(3), 246–251 (1975)
5. Choi, H.I., Han, C.Y.: Euler-Rodrigues frames on spatial Pythagorean-hodograph curves. Comput. Aided Geom. Design 19, 603–620 (2002)
6. Choi, H.I., Lee, D.S., Moon, H.P.: Clifford algebra, spin representation and rational parametrization of curves and surfaces. Adv. Comput. Math. 17, 5–48 (2002)
7. Farouki, R.T.: Exact rotation-minimizing frames for spatial Pythagorean-hodograph curves. Graphical Models 64, 382–395 (2002)
8. Farouki, R.T.: Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable. Springer, Berlin (2008)
9. Farouki, R.T., Giannelli, C., Manni, C., Sestini, A.: Quintic space curves with rational rotation-minimizing frames, Comput. Aided Geom. Design 26, 580–592 (2009)
10. Farouki, R.T., Sakkalis, T.: Pythagorean–hodograph space curves. Adv. Comput. Math. 2, 41–66 (1994)
11. Frankel, T.: The Geometry of Physics. Cambridge University Press, Cambridge (1999)
12. Han, C.Y.: Nonexistence of rational rotation-minimizing frames on cubic curves. Comput. Aided Geom. Design 6, 77–78 (2008)
13. Jüttler, B., Mäurer, C.: Cubic Pythagorean Hodograph Spline Curves and Applications to Sweep Surface Modeling. Comput. Aided Design 31, 73–83 (1999)
14. Jüttler, B.: Generating rational frames of space curves via Hermite interpolation with Pythagorean hodograph cubic splines. In: Geometric Modelling and Processing 1998, pp. 83–106. Bookplus Press, Soul (1998)
15. Leopoldseder, S.: Algorithms on cone spline surfaces and spatial osculating arc splines. Comput. Aided Geom. Design 18, 505–530 (2001)
16. Mäurer, C., Jüttler, B.: Rational approximation of rotation minimizing frames using Pythagorean-hodograph cubics. J. Geom. Graph. 3(2), 141–159 (1999)
17. Meek, D.S., Walton, D.J.: Geometric Hermite interpolation with Tschirnhausen cubics. J. Comput. Appl. Math. 81, 299–309 (1997)
18. Pottmann, H., Wagner, M.: Principal Surfaces. In: Goodman, T.N.T., Martin, R.R. (eds.) The Mathematics of Surfaces VII, Information Geometers, Winchester, pp. 337–362 (1997)
19. Pottmann, H., Wagner, M.: Contributions to Motion Based Surface Design. Int. J. of Shape Modeling 4, 183–196 (1998)
20. Ueda, K.: Spherical Pythagorean-Hodograph Curves. In: Dæhlen, M., Lyche, T., Schumaker, L.L. (eds.) Mathematical Methods for Curves and Surfaces II, pp. 485–492. Vanderbilt University Press, Nashville (1998)
21. Wagner, M., Ravani, B.: Curves with Rational Frenet–Serret motion. Comput. Aided Geom. Design 15, 79–101 (1997)
22. Wang, W., Joe, B.: Robust computation of the rotation minimizing frame for sweep surface modeling. Comput. Aided Design 29, 379–391 (1997)
23. Wang, W., Jüttler, B., Zheng, D., Liu, Y.: Computation of Rotation Minimizing Frame. ACM Trans. on Graphics 27(1), article no. 2 (2008)

Fat Arcs for Implicitly Defined Curves*

Szilvia Béla and Bert Jüttler

[1] Doctoral Program in Computational Mathematics
[2] Institute of Applied Geometry, Johannes Kepler University,
Altenberger Str. 69, 4040 Linz, Austria

Abstract. We present an algorithm generating a collection of fat arcs which bound the zero set of a given bivariate polynomial in Bernstein–Bézier representation. We demonstrate the performance of the algorithm (in particular the convergence rate) and we apply the results to the computation of intersection curves between implicitly defined algebraic surfaces and rational parametric surfaces.

1 Introduction

Bounding geometric primitives which enclose segments of planar curves are frequently needed for various geometric computations, e.g., for solving the intersection problem between two planar curves. Axis-aligned bounding boxes (min-max boxes), which can easily be generated both for planar parametric curves and for implicitly defined curves, are one of the simplest instances. Other useful primitives include fat lines (bounding strips, see e.g. [1]), the convex hull of the control polygon, or fat arcs [2].

The performance of a bounding primitive depends on the approximation order. For a bounding primitive with approximation order k, the number of primitives needed to bound a curve with a given tolerance ε grows like $\sqrt[k]{1/\varepsilon}$. Consequently, the use of geometric primitives with higher approximation order may provide computational advantages. Bounding boxes have only approximation order $k = 1$, while both the convex hull of control polygons and fat lines provide approximation order 2, and fat arcs even have approximation order 3.

Clearly, it is possible to define bounding primitives with an even higher approximation order. Fat arcs seems to be particularly useful since they provide a reasonable trade-off between geometry flexibility and the computational simplicity of elementary geometric operations. For instance, the computation of the intersection of two circular arcs requires solely the solution of quadratic equations, while this becomes far more complicated for higher order objects.

Various methods for generating an arc spline curve which approximate a given parametric curve with a prescribed tolerance have been described in the literature, see e.g. [3] for many related references. The use of arc splines for geometric design applications can be traced back to a classical VTO report of Sabin [4]. Marciniak and Putz dealt with the minimization of the number of arcs to approximate a curve under a give tolerance [5]. Later Qiu et al. improved their method [6]. In a number of papers, Meek and Walton applied arc splines to approximate parametric curves [7,8,9]

* This work was supported by the Austrian Science Found (FWF) through the Doctoral Program in Computational Mathematics, subproject 3.

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 26–40, 2010.

Yong used arc splines for quadratic Bézier curve approximation [10]. Feichtinger et al. compared various biarc interpolation schemes [11]. Held and Eibl approximated with biarcs simple planar polygons either for symmetric and asymmetric tolerance bounds [12].

Fat arcs as bounding geometric primitives were introduced by Sederberg [2]. Algorithms which generate bounding fat arcs for parametric curves are described, e.g., in [13]. Of course, any arc spline approximation technique can also be used to generate bounding fat arcs, simply by offsetting the obtained curve. The existing techniques for fat arc generation deal exclusively with the case of parametric curves.

In this paper we present an algorithm which generates a collection of fat arcs bounding an implicitly defined curve with a prescribed tolerance. First we describe how to find a fat arc for a single curve segment in a box. Then we combine this technique with adaptive subdivision in order to find a global approximation. As an application, we apply the fat arcs to approximate the intersection curve between implicitly defined and parametric surfaces.

2 Preliminaries

We recall the construction of fat arcs for parametric curves. In addition, we present a result concerning distance bounds and a criterion which guarantees single arcs.

2.1 Approximation by Arcs and Fat Arcs

In the case of planar parametric curves, the construction of fat arcs has been discussed in [13,2]. The methods described there generate an approximating arc which possesses a finite thickness. First, a median arc through three points of the original parametric curve segment is defined, see Fig. 1. For instance, these three points can be chosen as the two endpoints of the curve segment and the intersection point of the curve and the bisector of the endpoints. As the second step, the method computes (an upper bound of) the distance between the original curve segment and the median arc. Finally, an offset of the

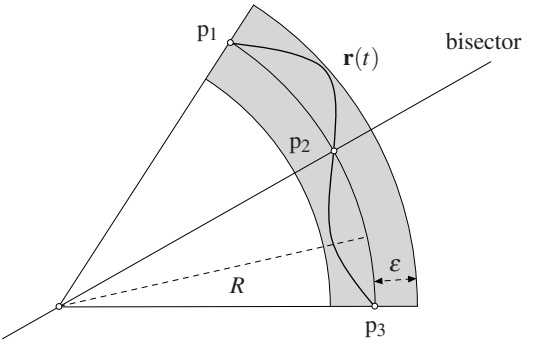


Fig. 1. Fat arc generation

median with this distance is defined. The boundaries of the offset are concentric arcs, whose radii are the sum and the difference of the median arc radius and the distance of the curves. They define a bounding domain for the original curve segment. This bounding region is called a *fat arc*. Since the approximation order of circular arcs is equal to three, the offset distance behaves as $O(h^3)$, where h is the length of the given curve segment (or of its parameter interval).

2.2 Distance Estimate

In order to construct fat arcs for an implicitly defined curve, we shall use the properties of the defining function. We assume that the bivariate polynomial is given by its tensor-product Bernstein-Bézier (BB) representation of degree (m,n):

$$f(\mathbf{x}) = \sum_{i=0}^{m} \sum_{j=0}^{n} d_{ij} \beta_i^m(x_1) \beta_j^n(x_2), \tag{1}$$

with respect to the domain $\Omega = [0,1]^2$ and with certain coefficients $d_{ij} \in \mathbb{R}$, where

$$\beta_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad t \in [0,1]. \tag{2}$$

More generally, (x_1, x_2) will also be used as local coordinates

$$x_1 = \frac{\xi_1 - a_1}{h}, \quad x_2 = \frac{\xi_2 - a_2}{h}, \tag{3}$$

with respect to the general axis-aligned box $B = [a_1, a_1 + h] \times [a_2, a_2 + h]$ of size $h \times h$ in the (ξ_1, ξ_2)-plane.

The implicitly defined curve is given as the zero set of the bivariate polynomial,

$$\mathscr{F} = \{\mathbf{x} : f(\mathbf{x}) = 0 \wedge \mathbf{x} \in [0,1]^2\}. \tag{4}$$

Clearly, the curve may be empty, or it may consist of more than one curve segment.

In order to construct a fat arc, we need to bound the distance between the median arc and the curve using a result from [14], see also [15]. On the one hand, we consider the medial arc as a parametric curve $\mathbf{g} : t \mapsto \mathbf{g}(\mathbf{t})$ with parameter domain $t \in [a,b]$, which traces the point set

$$\mathscr{G} = \{\mathbf{g}(t) : t \in [0,1]\} \tag{5}$$

where we assume that $\mathscr{G} \subset [0,1]^2$. On the other hand, in order to avoid certain technical difficulties, we consider the set

$$\mathscr{F}^* = \mathscr{F} \cup \partial\Omega \tag{6}$$

which is obtained by adding the boundary of the domain to the curve $\mathscr{F}$. The one-sided Hausdorff distance of $\mathscr{F}^*$ and $\mathscr{G}$ is defined as

$$\mathrm{HD}(\mathscr{G}, \mathscr{F}^*) = \sup_{t \in [0,1]} \inf_{\mathbf{x} \in \mathscr{F}^*} \|\mathbf{x} - \mathbf{g}(t)\|. \tag{7}$$

We recall the following result from [14]:

Lemma 1. *If there exist positive constants c, η such that*

$$\forall \mathbf{x} \in \Omega: \quad c \le \|(\nabla f)(\mathbf{x})\| \quad \text{and} \quad \forall t \in [0,1]: \quad |(f \circ \mathbf{g})(t)| \le \eta \tag{8}$$

hold, then the one-sided Hausdorff distance is bounded by

$$\mathrm{HD}(\mathscr{G}, \mathscr{F}^*) \le \frac{\eta}{c}. \tag{9}$$

Consequently, the parametric curve is contained in ε–neighborhood of $\mathscr{F}^*$, where $\varepsilon = \eta/c$. However, it should be noted that this distance bound does not guarantee that the implicitly defined curve is then also contained in an ε–neighborhood of the parametric curve. The algorithm presented below uses an additional test to guarantee this property. Nevertheless, in all computed examples the above distance bound provided a safe and conservative estimate for the two-sided Hausdorff distance of the implicitly defined and the parametric curve.

2.3 Evaluation of the Constants

In order to find the constants c and η in Lemma 1, we represent the median arc as a quadratic rational Bézier curve,

$$\mathbf{c}(t) = \sum_{i=0}^{2} \mathbf{c}_i \frac{\tilde{w}_i \beta_i^2(t)}{\sum_{j=0}^{2} \tilde{w}_j \beta_j^2(t)}, \quad t \in [0,1]. \tag{10}$$

The composition $f \circ \mathbf{c}$ is a rational function of degree $2(m+n)$ which can be represented by its BB representation with certain coefficients d_i and weights w_i. The weights are found by evaluating the $(m+n)$th power of the denominator in (10). If all weights and coefficients are positive, then

$$|(f \circ \mathbf{c})(t)| = \sum_{i=0}^{2m+2n} \frac{d_i w_i \beta_i^{2n+2m}(t)}{\sum_{j=0}^{2m+2n} w_j \beta_j^{2m+2n}(t)} \le \frac{\max\limits_i d_i w_i}{\min\limits_i w_i} = \eta. \tag{11}$$

In order to find the second constant c, we generate the tensor–product BB representation

$$\left(\frac{\partial f(x_1,x_2)}{\partial x_1}\right)^2 + \left(\frac{\partial f(x_1,x_2)}{\partial x_2}\right)^2 = \sum_{i=0}^{2m} \sum_{j=0}^{2n} h_{ij} \beta_i^{2m}(x_1) \beta_j^{2n}(x_2) \tag{12}$$

which can be found using differentiation, product, and degree elevation formulas, see [16]. If all coefficients h_{ij} are positive, then

$$\|\nabla f(\mathbf{x})\| \ge \sqrt{\min_{i,j} h_{ij}} = c. \tag{13}$$

2.4 Identifying Boxes with Single Segments

In order to generate a fat arc approximation in a box, we need to ensure that the box contains only one segment of the implicitly defined curve segment. Various criteria for

isolating a regular segment of an algebraic curve have been discussed in the literature. For instance, different types of discriminating curve families have been used in [17]. These discriminating families are particularly useful in combination with algorithms that trace the algebraic curve segments.

In our case, we are interested in a criterion which guarantees that the box Ω contains a single curve segment with exactly two intersections with the boundaries.

Lemma 2. *Consider an algebraic curve segment defined by the polynomial (1). We say that the coefficients exhibit a* ***corner event****, if*

- *the coefficient at one of the corners is equal to zero and*
- *the first non-zero coefficients along the two neighboring boundaries have a different sign.*

We say that the the coefficients exhibit an ***edge event****, if*

- *the control polygon along one the box boundaries has exactly one sign change from plus to minus or vice versa.*

If the number of the corner and edge events is equal to two and if the positive constant c of Lemma 1 exists, then the box contains a single curve segment, which is regular, connected, and which intersects the box boundary in exactly two points.

For the proof it suffices to observe that each event guarantees that the implicitly defined curve crosses the box boundary in exactly one point. Moreover, the curve does not contain any closed loops, since the gradient vector does not vanish in Ω.

The conditions of Lemma 2 are sufficient, but not necessary. For example, the lemma excludes the case of a single arc of the implicitly defined curve crossing the same edge twice.

3 Approximation of a Single Curve Segment

This section focuses on the approximation of a single segment of the algebraic curve. We describe the algorithm and demonstrate its performance.

3.1 Outline

The algorithm `FatArcSegment` (see Alg. 1) is based on the corresponding techniques in the parametric case, but it uses the bounds which are obtained with the help of Lemma 1. It assumes that the conditions of Lemma 2 are satisfied.

The algorithm is successful, if it finds three points $\mathbf{b}_1, \mathbf{b}_2, \mathbf{c}$ of the median arc, the fat arc thickness is smaller then the prescribed tolerance ε and if there is no sign change of f along the boundaries of the fat arc. It the returns the fat arc which bounds the curve segment.

It may happen, that there are no fat arc boundaries, or only one of the bounding arcs can be generated (e.g. when the distance bound of the median arc and the implicitly defined curve is greater then the radius of the meridian circle, or one of the bounding

Algorithm 1. `FatArcSegment` (f, B, ε)

Require: The conditions of Lemma 2 are satisfied.

1: $b = \{\mathbf{b}_1, \mathbf{b}_2\} \leftarrow$ approximate boundary points of the implicitly defined curve
2: $c = \{\mathbf{c}\} \leftarrow$ approximate inner point of the implicitly defined curve
3: **if** $\#b = 2$ and $\#c = 1$ **then**
4: $\quad \mathscr{M} \leftarrow$ circle through $\mathbf{b}_1, \mathbf{c}, \mathbf{b}_2$ {median circle}
5: $\quad d \leftarrow$ upper bound of $\mathrm{HD}(\mathscr{M} \cap B, \mathscr{F}^*)$ {see Lemma 1}
6: $\quad$ **if** $d \leqslant \varepsilon$ and $d \leqslant$ radius of $\mathscr{M}$ **then**
7: $\quad\quad \mathscr{C}_d \leftarrow$ offset ring of $\mathscr{M}$ with distance d {fat circle}
8: $\quad\quad \mathscr{C}^+, \mathscr{C}^- \leftarrow$ inner and outer circle of $\partial \mathscr{C}_d$
9: $\quad\quad$ **if** there is no sign change of f along $\mathscr{C}^+ \cap B$ or $\mathscr{C}^- \cap B$ **then**
10: $\quad\quad\quad$ **return** $B \cap \mathscr{C}_d$ {fat arc has been found}
11: $\quad\quad$ **end if**
12: $\quad$ **end if**
13: **end if**
14: **return** $\emptyset$ {no fat arc has been found}

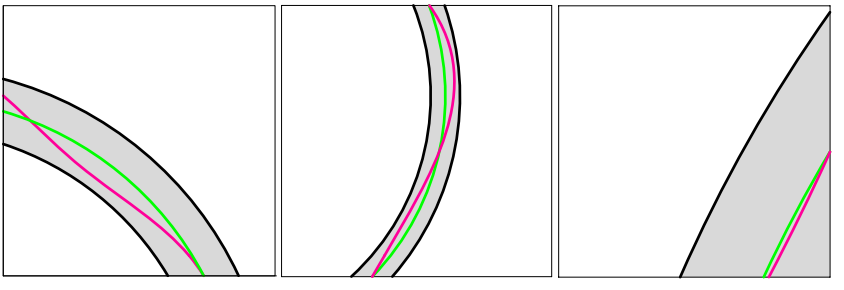


Fig. 2. Examples for fat arc generation with the help of algorithm `FatArcSegment`. The red curves are the implicitly defined curves. The median circles are shown in green.

arcs does not intersect the box). The local algorithm fails if no fat arcs are generated. The algorithm then returns the empty set.

Figure 2 presents three examples of fat arcs which have been generated with the help of the algorithm. The individual steps of the algorithm are described in the next section.

3.2 Details

We describe the generation of approximate points of the curve (lines 1 and 2 of the algorithm), the segmentation of the fat arc boundaries and the sign change analysis along the segmented boundaries (line 9).

Approximate Intersection Points: In order to construct the median arc (line 5), we approximate three points of the implicitly defined curve. Two of them are the intersections with the boundaries of the box, while the third point is the intersection with the bisector of the first two points.

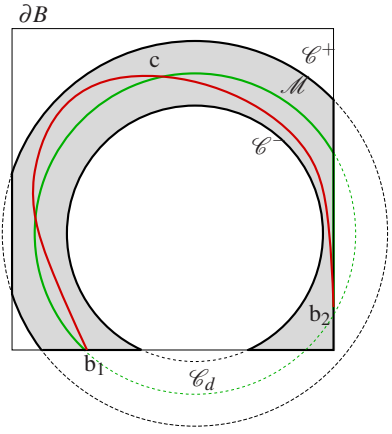


Fig. 3. Fat arc segmentation with a box: the thick black curves are the segmented fat arc boundaries

In the case of a corner event, the corner is a boundary point of the curve. In the case of an edge event, the corresponding edge contains an intersection of the curve with the boundary of the box. It is then approximated as follows: We consider the restriction of f to the edge and generate its best L^2 approximation by a quadratic polynomial q^* which additionally interpolates the values of f at the two neighboring corners. The root of q^* then defines the approximate intersection of $\mathscr{F}$ with the edge. If no simultaneous corner event occurs at the neighbouring two corner points, then there is exactly one root, since the BB coefficients of f possess exactly one sign change from plus to minus or vice versa.

After generating the first two points, we restrict the function f to the intersection of the bisector with the box. Again we generate its best L^2 approximation by a quadratic polynomial q^* which additionally interpolates the values of f at the two end points. The root of q^* then defines the approximate intersection of $\mathscr{F}$ with the bisector.

Using similar arguments as in [18] it can be shown that the distance of the approximate points to the curve behaves as $\mathscr{O}(h^3)$, where h is the size of the box in global coordinates, cf. Eq. (3).

Analysis of Fat Arc Boundaries: The fat arc in line 10 of the algorithm is obtained by intersecting the circular ring $\mathscr{C}_d$ with the box. Its boundaries consists of segments of circular arcs and of segments of box boundaries. See Fig. 3 for several examples. We create the segments of the circular arcs and represent them by rational quadratic Bézier curves. These segments will be referred to as the boundaries of the fat arc.

Since the diameter d of the fat arc is only a bound on the one–sided Hausdorff distance $\mathrm{HD}(\mathscr{G}, \mathscr{F}^*)$, we need to verify that the fat arc indeed contains the implicitly defined curve segment. This is done by analyzing the sign of f along the fat arc boundaries.

For each segment of the fat arc boundary, we compose f with the quadratic rational parametrization and obtain a rational function in Bernstein-Bézier form. If all weights and all coefficients of the numerator have the same sign, respectively, then no sign changes of f are present (see line 9 of the algorithm).

3.3 Approximation Order

Since the approximation order of curves by segments of circular arcs is three (see [2]), the same result is anticipated for the results produced by algorithm `FatArcSegment`. We confirm this approximation order by numerical examples. Consider the three bivariate polynomials

$$\begin{aligned} f_1(\mathbf{x}) &= x_1^4 + x_1^3 x_2^2 + 2x_1^2 x_2 - 6x_1 x_2 + x_2^4 - 8x_2^2 - 12x_2 \\ f_2(\mathbf{x}) &= -x_1^3 - x_1^2 x_2 + x_1 x_2 - x_2^3 + x_2^2 - 2x_2 \\ f_3(\mathbf{x}) &= -4x_1^3 - 5x_1^2 + 2x_2 \end{aligned} \tag{14}$$

with the domains (in global coordinates)

$$B^k = [-10^{-k}, 10^{-k}] \times [-10^{-k}, 10^{-k}], \quad k \in \mathbb{R}. \tag{15}$$

In the case of the first polynomial, Fig. 4 shows the result of the fat arc constructions for several values of k. The implicitly defined curve is the red one, the median arc denoted with green and the fat arcs are represented with black.

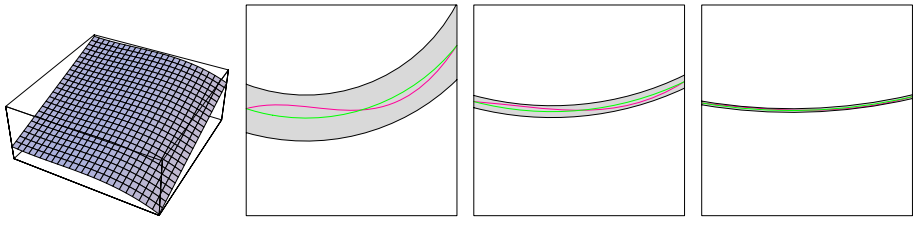


Fig. 4. Left: The graph of f_1. Right: Fat arcs for $k = 0.5, 0.75, 1.0$.

Fig. 5 visualizes the relation between the width of the fat arc and the size of the box for the three polynomials in (14). For sufficiently large values of k, the slopes of the three curves in the doubly-logarithmic plot are all three, thus confirming the expected approximation order.

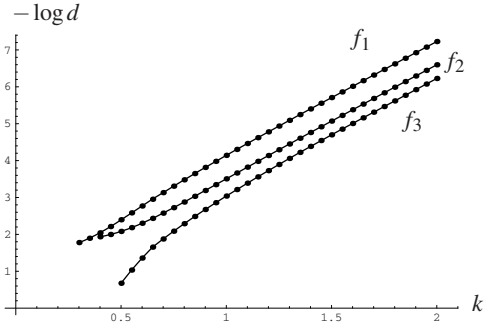


Fig. 5. Dependency between diameter and box size

4 Approximation of an Implicitly Defined Curve

We describe an algorithm which generates a collection of fat arcs bounding a given implicitly defined curve in a box. We present several examples and compare the performance of fat arcs with bounding boxes.

4.1 The Global Algorithm

The algorithm `GenerateFatArcs` (see Alg. 2) combines `FatArcSegment` with recursive subdivision. First it analyzes the signs of the Bernstein–Bézier coefficients with respect to the current box. If no sign changes are present, then the current box does not contain any components of the implicitly defined curve. Otherwise it tries to apply the algorithm for a single segment. If this is not successful, then the algorithm either subdivides the current box into four squares or returns the entire box, if its diameter is already below the user-defined threshold ε.

Algorithm 2. `GenerateFatArcs`(f, B, ε)

```
1:  if min d_ij > 0 or max d_ij < 0 then
2:     return ∅                                          {the box is empty}
3:  end if
4:  if f satisfies the assumptions of Lemma 2 then
5:     A ← FatArcSegment(f, B, ε)                        {single fat arc generation}
6:     if A ≠ ∅ then
7:        return A                                       {... has been successful}
8:     end if
9:  end if
10: if diameter of B > ε then
11:    subdivide the box into 4 subboxes B_1, ..., B_4   {quadsection}
12:    return ⋃_{i=1}^{4} GenerateFatArcs(f, B_i, ε)     {recursive call}
13: end if
14: return B                                             {current box is small enough}
```

Note that the algorithm may return boxes which do not contain any segments of the implicitly defined curve ("false positive boxes"). However, it is guaranteed that it returns a region which contains the implicitly defined curve.

4.2 Examples

We illustrate the performance of the algorithm by four examples.

Example 3. The first example (see Fig. 6) visualizes the entire algorithm. We apply the algorithm to a bivariate polynomial of degree $(1,4)$ which has only one arc in the region of interest and choose a relatively large tolerance ε. The first call of the algorithm produces four subboxes which are then analyzed independently. The first box contains

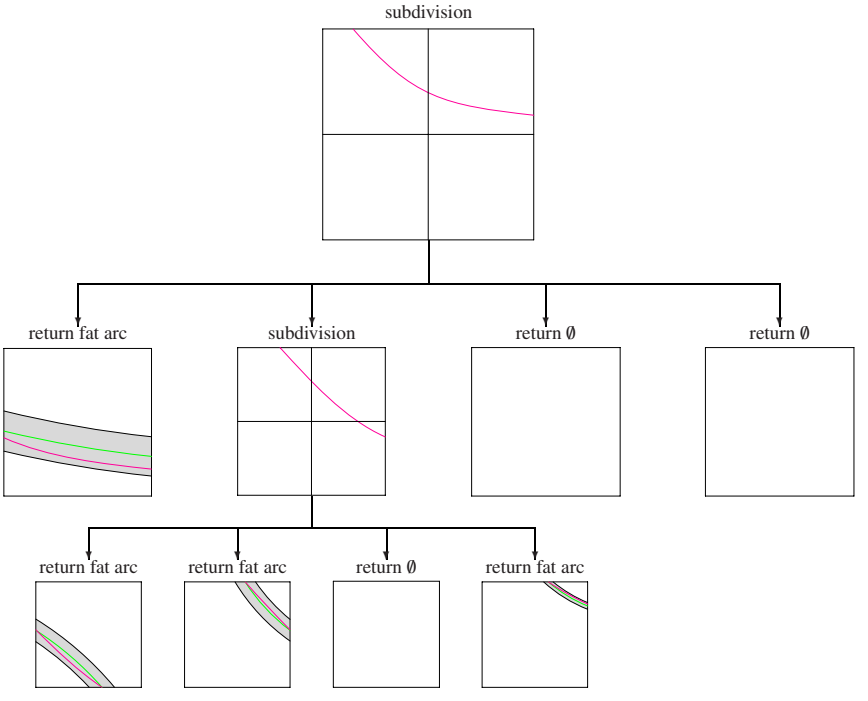


Fig. 6. Example 3: The decision tree of algorithm `GenerateFatArcs`

an arc which can be approximated by a single fat arc. The second box produces other four subboxes, while the third and the fourth boxes do not contain any points of the implicitly defined curve. Finally, analyzing the four second-generation subboxes leads to three additional fat arcs and one empty box. The output is generated by collecting all boxes in the leaves of the tree.

Example 4. We consider a polynomial f of degree (6,9) with randomly generated BB coefficients in $[-1, 1]$. Figure 7(a) shows the surface and the implicitly defined curve segments. Figure 7(b) and (c) demonstrate the behavior of the algorithm for different tolerances ε. The upper row shows the entire domain, while the lower row shows a zoomed view of the lower left corner of the box. In the case of $\varepsilon = 0.1$, which is shown in (b), some boxes are returned as bounding boxes, since `FatArcSegment` fails and the diameter of the boxes is smaller than ε. For the smaller value of $\varepsilon = 0.01$, the fat arc generation succeeded in all generated boxes.

In the next two examples we compare fat arcs with (recursively generated) bounding boxes. In the latter case we accepted boxes with a diameter less than the prescribed tolerance.

Example 5. We approximate an implicitly defined curve, see Figure 8, by fat arcs (a) and by bounding boxes (b). Clearly, the use of fat arcs leads to a much smaller number

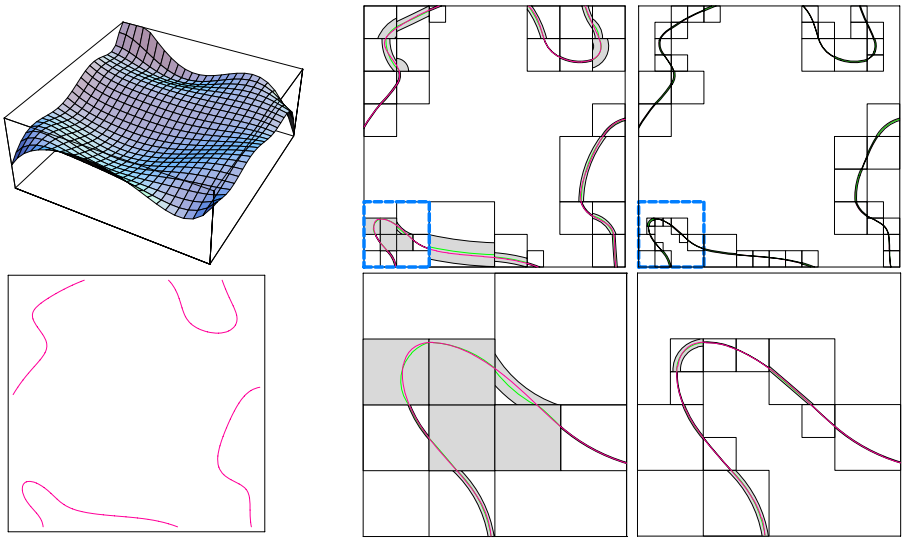


Fig. 7. Example 4: Fat arc generation for different tolerances. The graph of f and the implicitly defined curve (a), and The fat arcs (top) and a zoomed view (bottom) for $\varepsilon = 0.1$ (b) and for $\varepsilon = 0.01$ (c).

of bounding geometric primitives. This becomes even more dramatic for smaller tolerances. The plot in (c) shows the relation between the number of generated primitives (fat arcs or boxes) and the tolerance $\varepsilon = \sqrt{2}/2^k$.

Example 6. This example is based on an implicitly defined curve which possesses a singular point, see Figure 9. In this situation, the fat arc generation will fail for any box which contains the singular point, since no positive lower bound c on $||\nabla f||$ exists. Consequently, the algorithm always returns a box containing this point. Still, the results generated by our method (left) compare favorably with the use of bounding boxes (right).

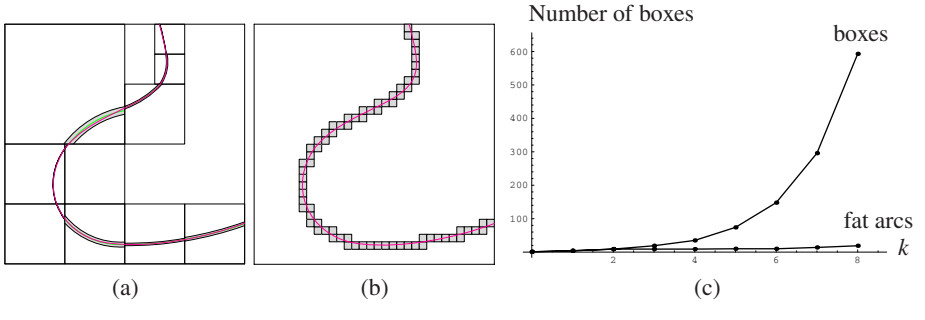


Fig. 8. Example 5: Comparison of fat arcs (a) and bounding boxes (b). The relation between tolerance and number of bounding primitives (c).

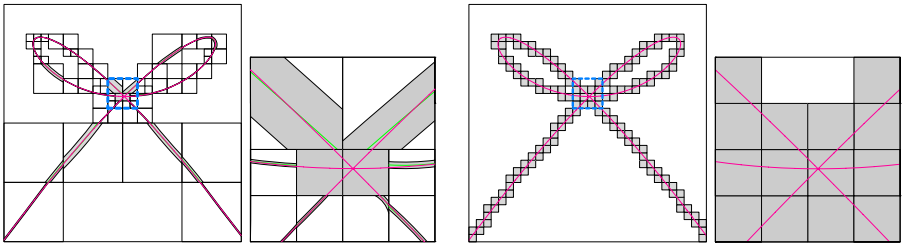


Fig. 9. Example 6: Fat arcs (left) and bounding boxed (right) for an implicitly defined curve with a singular point, where $\varepsilon = \sqrt{2}/2^5$

5 Approximation of Surface-Surface Intersections

As an application we apply the fat arc generation algorithm to the intersection problem between implicitly and parametrically defined surfaces. The computation of surface-surface intersections is a potential application of bounding region generation methods. In practice, intersection computation of a parametric and an implicitly defined surface is one of the most frequently encountered cases [19]. A good survey is given by [20,21] in this topic.

Consider an implicitly defined surface $h(x,y,z) = 0$ and a parametric surface patch $\mathbf{r}(\xi_1,\xi_2)$ with domain $\Omega = [0,1]^2$. Then the implicitly defined curve $f = h \circ r = 0$ describes the intersection curve in the domain of the parametric surface patch.

Using Algorithm `GenerateFatArcs` one may now construct a collection of fat arcs with maximum width ε in Ω. The region described by them corresponds to a certain subset (a strip) on the parametric surface patch.

Recall that the coefficients of the first fundamental form are defined as

$$g_{ij}(\xi_1,\xi_2) = \frac{\partial}{\partial \xi_i}\mathbf{r}(\xi_1,\xi_2) \cdot \frac{\partial}{\partial \xi_j}\mathbf{r}(\xi_1,\xi_2). \tag{16}$$

In order to relate the thickness of the bounding fat arcs to the thickness of the corresponding strip on the parametric surface, we present the following observation.

Lemma 7. *We assume that there exists a constant C such that*

$$g_{11}(\xi_1,\xi_2)\cos^2\phi + 2g_{12}(\xi_1,\xi_2)\cos\phi\sin\phi + g_{22}(\xi_1,\xi_2)\sin^2\phi \le C \tag{17}$$

holds for all $\phi \in [0,2\pi]$ and $(\xi_1,\xi_2) \in \Omega^0$. Consider a single fat arc with width ε in the parameter domain. Then the width of the corresponding region on the parametric surface patch is bounded by $2\varepsilon\sqrt{C}$.

Proof. The length L of a curve on the surface which corresponds to any straight line segment

$$(\xi_1(t),\xi_2(t)) = (\xi_1^0,\xi_2^0) + t(\cos\phi,\sin\phi), \quad t \in [a,b] \tag{18}$$

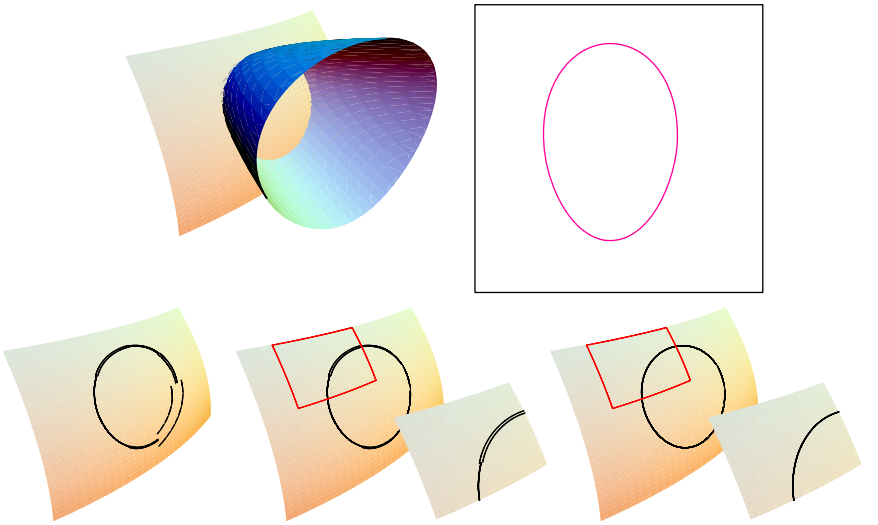


Fig. 10. Example 8: Intersection of a cubic implicit and a biquadratic parametric surface, represented by fat arcs in the parameter domain. The number of fat arcs grows from 10 for $\varepsilon = 0.1$ to 25 for $\varepsilon = 0.01$. For the larger two tolerances, we also zoomed into a segment of the surface patch.

in the parameter domain satisfies

$$L = \int_a^b \sqrt{g_{11}\cos^2\phi + 2g_{12}\cos\phi\sin\phi + g_{22}\sin^2\phi}\,\mathrm{d}t \le (b-a)\sqrt{C}. \qquad (19)$$

This observation can now be applied to the lines which pass through the center of the fat arc.

If $\mathbf{r}$ is a rational surface patch, then the constant C in (17) can be found by found by replacing $(\cos\phi, \sin\phi)$ by a rational parametrization of a semicircle[1], say with a parameter $\tau \in [0,1]$. Then the left–hand side of (17) defines a trivariate rational function which depends on the three parameters $(\xi_1, \xi_2, \tau) \in [0,1]^3$ and the bound C can be found with the help of the rational BB representation.

Example 8. We consider the intersection of a cubic implicitly defined surface with a biquadratic surface patch. Fig. 10, upper row, shows the intersecting surfaces and the implicitly defined intersection curve in the parameter domain. The lower row shows the region on the surface which correspond to fat arcs in the parameter domain for three different values of the tolerance ε.

[1] A semicircle is sufficient, due to the symmetry of the quadratic form in (17).

6 Conclusion

We have presented an algorithm which generate a collection of bounding fat arcs for a given planar curve. In contrast to the existing techniques, which mostly assume that parametric representations are given, the algorithm can be applied to implicitly defined curves.

The planned future work includes the extension to surfaces and to space curves, which can be bounded by toroidal surface segments. We also plan to use the results for solving systems of polynomial equations, where bounding primitives of higher approximation order may help to accelerate the convergence [18,22]. Finally it might also be interesting to apply the algorithm to the problem of analyzing and certifying the topology of an algebraic curve, see e.g. [23].

As another possible extension, one might consider other types of median curves, such as conic sections or algebraic curves of degree higher than two. While this may provide an even higher rate of convergence (cf. [24]), it makes the generation of the fat curve more difficult (since e.g. the class of conic sections, unlike circular arcs, is not closed under offsetting). As an advantage, the use of circular arcs leads to simpler algorithms for intersection computation.

References

1. Ballard, D.H.: Strip trees: a hierarchical representation for curves. Commun. ACM 24(5), 310–321 (1981)
2. Sederberg, T.W., White, S.C., Zundel, A.K.: Fat arcs: a bounding region with cubic convergence. Comput. Aided Geom. Des. 6(3), 205–218 (1989)
3. Yang, X.: Efficient circular arc interpolation based on active tolerance control. Computer-Aided Design 34(13), 1037–1046 (2002)
4. Sabin, M.A.: The use of circular arcs for the interpolation of curves through empirical data points. Technical Report VTO/MS/164, British Aircraft Corporation (1976)
5. Marciniak, K., Putz, B.: Approximation of spirals by piecewise curves of fewest circular arc segments. Computer-Aided Design 16(2), 87–90 (1984)
6. Qiu, H., Cheng, K., Li, Y.: Optimal circular arc interpolation for NC tool path generation in curve contour manufacturing. Computer-Aided Design 29(11), 751–760 (1997)
7. Walton, D.S., Meek, D.J.: Approximating quadratic NURBS curves by arc splines. Computer-Aided Design 25(6), 371–376 (1993)
8. Walton, D.S., Meek, D.J.: An arc spline approximation to a clothoid. Journal of Computational and Applied Mathematics 170(1), 59–77 (2004)
9. Walton, D.S., Meek, D.J.: Spiral arc spline approximation to a planar spiral. Journal of Computational and Applied Mathematics 170(1), 59–77 (2004)
10. Yong, J.H., Hu, S.M., Sun, J.G.: Bisection algorithms for approximating quadratic bézier curves by G^1 arc splines. Computer-Aided Design 32(4), 253–260 (2000)
11. Šír, Z., Feichtinger, R., Jüttler, B.: Approximating curves and their offsets using biarcs and Pythagorean hodograph quintics. Comp.–Aided Design 38, 608–618 (2006)
12. Held, M., Eibl, J.: Biarc approximation of polygons within asymmetric tolerance bands. Computer-Aided Design 37(4), 357–371 (2004)
13. Lin, Q., Rokne, J.G.: Approximation by fat arcs and fat biarcs. Computer-Aided Design 34(13), 969–979 (2002)

14. Aigner, M., Szilagyi, I., Schicho, J., Jüttler, B.: Implicitization and distance bounds. In: Mourrain, B., Elkadi, M., Piene, R. (eds.) Algebraic Geometry and Geometric Modeling, pp. 71–86. Springer, Heidelberg (2006)
15. Sederberg, T.W.: Planar piecewise algebraic curves. Comp. Aided Geom. Design 1, 241–255 (1984)
16. Hoschek, J., Lasser, D.: Fundamentals of Computer Aided Geometric Design. AK Peters, Wellesley (1993)
17. Xu, G., Bajaj, C.L., Xue, W.: Regular algebraic curve segments (i) – definitions and characteristics. Comput. Aided Geom. Des. 17(6), 485–501 (2000)
18. Bartoň, M., Jüttler, B.: Computing roots of polynomials by quadratic clipping. Comp. Aided Geom. Design 24, 125–141 (2007)
19. Lee, K.: Principles of CAD/CAM/CAE Systems. Addison-Wesley, Reading (1999)
20. Pratt, M.J., Geisow, A.D.: Surface/surface intersection problem. In: Gregory, J. (ed.) The Mathematics of Surfaces II, pp. 117–142. Claredon Press, Oxford (1986)
21. Patrikalakis, N.M., Maekawa, T.: Shape interrogation for computer aided design and manufacturing. Springer, Heidelberg (2002)
22. Mourrain, B., Pavone, J.P.: Subdivision methods for solving polynomial equations. Journal of Symbolic Computation (2008) (accepted manuscript)
23. Gonzalez-Vega, L., Necula, I.: Efficient topology determination of implicitly defined algebraic plane curves. Comput. Aided Geom. Design, 719–743 (2002)
24. Dokken, T.: Approximate implicitization. In: Lyche, T., Schumaker, L. (eds.) Mathematical methods for curves and surfaces, pp. 81–102. Vanderbilt University Press, Nashville (2001)

Geometric Properties of the Adaptive Delaunay Tessellation

Tom Bobach[1], Alexandru Constantiniu[2], Paul Steinmann[2], and Georg Umlauf[1]

[1] Dept. of Computer Science, University Kaiserslautern, Germany
{bobach,umlauf}@informatik.uni-kl.de
[2] Institute of Applied Mechanics, University of Erlangen-Nuremberg, Germany
{constantiniu,paul.steinmann}@ltm.uni-erlangen.de

Abstract. Recently, the *Adaptive Delaunay Tessellation* (ADT) was introduced in the context of computational mechanics as a tool to support Voronoi-based nodal integration schemes in the finite element method. While focusing on applications in mechanical engineering, the former presentation lacked rigorous proofs for the claimed geometric properties of the ADT necessary for the computation of the nodal integration scheme. This paper gives pending proofs for the three main claims which are uniqueness of the ADT, connectedness of the ADT, and coverage of the Voronoi tiles by adjacent ADT tiles. Furthermore, this paper provides a critical assessment of the ADT for arbitrary point sets.

1 Introduction

One building block in finite element analysis is the background tessellation of a spatial domain, where requirements vary with the applications at hand. The corresponding field of mesh generation traditionally focuses on the generation of both vertex positions and connectivity, starting from a description of the domain boundary. Vertices are placed heuristically to adaptively sample local features and maintain a good shape of the resulting elements, although the discussion about the optimal shape is not settled even for triangles [1]. The tessellation can be done in ad-hoc constructions [2], iteratively [3], or based on guidance fields [4,5] to name a few. In the rare occasion of a predetermined set of vertices, the choice of methods is generally limited to the constrained Delaunay triangulation or other, data-dependent, triangulations [6].

In [7], a novel tessellation technique, called *Adaptive Delaunay Tessellation* (ADT), was introduced in the context of computational mechanics as a tool to support Voronoi-based nodal integration schemes in the Finite Element Method. Its main contribution is a simplified access to the element structure that supports nodal integration schemes based on a robust and unique transformation of the Delaunay triangulation. While focusing on the applications in linear elasticity, the presentation in [7] lacks rigorous proofs for the geometric properties of the ADT which are uniqueness of the ADT, connectedness of the ADT, and coverage of the Voronoi tiles by adjacent ADT tiles. These properties are essential for the computation of the nodal integration scheme. The pending proofs for these main claims are given in this paper.

We start with a motivation for the ADT and introduce some notation used in the rest of this paper in Section 2. Uniqueness and connectedness of the ADT are stated and

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 41–54, 2010.

proved in Sections 3.1 and 3.2. Section 3.3 introduces an alternative characterization of the ADT to simplify the proof of the last important property in Section 3.4 on the coverage of the Voronoi tile of a vertex by ADT elements adjacent to it. We discuss the ADT in Section 4 and conclude in Section 5.

1.1 Linear Elasticity and Nodal Integration in the Finite Element Method

The *Finite Element Method* encompasses a wide range of applications to model complex continuous phenomena based on a description composed of many small parts, called finite elements, which allow a structurally simpler definition of the phenomenon. In the field of computational mechanics, which is the main target of the ADT, one of the most basic and well-studied problems is that of linear elasticity, from which we sketch the key aspects in a simplified form.

The Finite Element Method typically requires the evaluation of the integral

$$\int_{\Omega} \nabla \psi \cdot f$$

for the so-called *test function* ψ, and some function f. In general, ψ is an interpolant that is defined piecewise over the elements Ω_i of the domain. This allows the decomposition of the global integral into a sum of local integrals,

$$\sum_i \int_{\Omega_i} \nabla \psi \cdot f.$$

This is the heart of the Finite Element Method and allows the numerical assessment of the problem by concentrating on its local reformulation for each Ω_i and the associated local interpolant $\psi_i = \psi\Big|_{\Omega_i}$. Disadvantages of the element-based approach are the requirement of derivatives of the test function ψ, and numerical problems if the elements Ω_i are badly shaped, as discussed in [1].

One remedy for these problems is to turn to meshless methods, for which the discretization focuses on vertices rather than elements. One meshless approach based on natural neighbor coordinates has been proposed in [8]. Another meshless technique is *nodal integration*, as introduced in [9]. Here, an alternative tessellation of the domain is considered, where the vertices $\mathbf{v}_j$ are enclosed by polygons $\mathscr{T}_j$, with $\Omega = \bigcup_j \mathscr{T}_j$ and the above integral becomes

$$\sum_j \int_{\mathscr{T}_j} \nabla \psi \cdot f,$$

assigning a part of the integral to each node $\mathbf{v}_j$ rather than to an element Ω_i. The canonical choice for this tessellation is the Voronoi diagram of the vertices $\mathbf{v}_j$ of the background tessellation, restricted to Ω. The application of Stokes' theorem allows to write

$$\int_{\mathscr{T}_j} \nabla \psi \cdot f = \int_{\partial \mathscr{T}_j} \mathbf{n} \cdot \psi \cdot f,$$

where $\mathbf{n}$ is the outward pointing unit vector. By turning the area integral into a path integral, no derivatives of ψ are required, but now the local integration domains $\mathscr{T}_i$ do no longer correspond to the elements Ω_j of the background tessellation. Therefore, the computation of the path integral for each node can no longer be carried out on a per-element basis, but requires the evaluation of the test function ψ at arbitrary positions along $\partial\mathscr{T}_j$.

1.2 Voronoi Tile Coverage in the ADT

By covering the Voronoi tile of a vertex with a minimal set of adjacent polygons, the computational assessment of Voronoi-based nodal integration is greatly simplified. Figure 1 shows a set of scattered points covering a rectangular domain. The boundary of the Voronoi tile shown in Figure 1(b) is a convex polygon that intersects certain triangles in the Delaunay triangulation of the point set, which are not necessarily adjacent to the Voronoi site. The ADT, on the other hand, consists of polygons that are big enough to cover the whole Voronoi tile. Thus, by traversing the polygons adjacent to a vertex, one has access to an area that completely covers the Voronoi tile of that vertex.

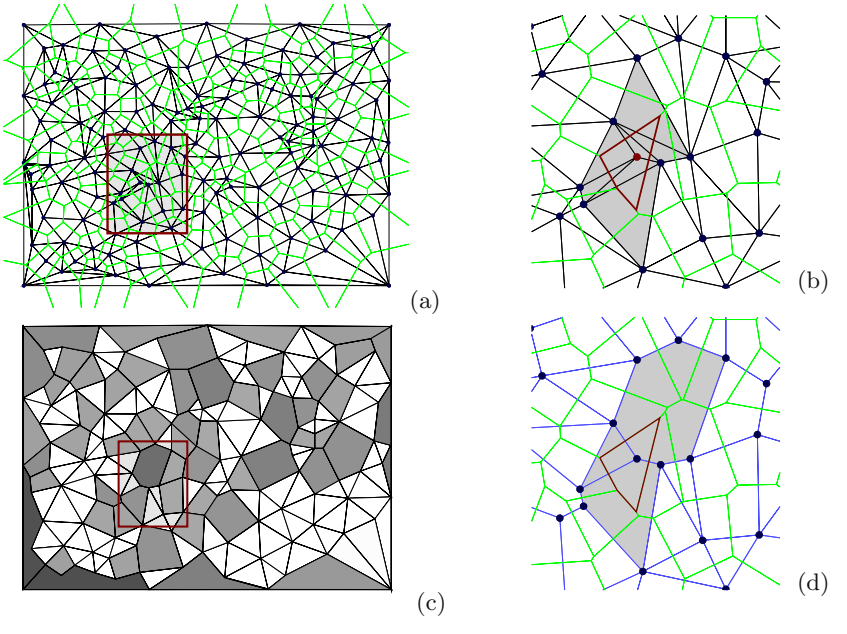


Fig. 1. Example of an ADT tessellation. The point cloud with its Voronoi diagram in green and the Delaunay triangulation in black is shown in (a). The closeup of a node with shaded triangles depicting the necessary cover for its Voronoi tile is shown in (b). The corresponding ADT with polygons shaded according to valence is shown in (c). In the closeup (d), it is apparent that the adjacent ADT polygons cover the Voronoi tile.

2 The Adaptive Delaunay Tessellation

Before presenting the main findings, we introduce the necessary notation and repeat some important results.

We consider the *canonical open ball topology* on $\mathbb{R}^2$ and denote by $[\Omega]$ the *topological closure* of a set Ω, by $]\Omega[$ its *topological interior*, and by $\partial\Omega := [\Omega]\backslash]\Omega[$ its *topological boundary*. For a set $\mathbf{P} \subset \mathbb{R}^2$ of points, we denote by $\mathscr{C}(\mathbf{P}) =]\mathscr{C}(\mathbf{P})[\cup \partial\mathscr{C}(\mathbf{P})$ the convex hull of $\mathbf{P}$.

We denote by (F,E) a partition of $\mathscr{C}(\mathbf{P})$ into polygonal faces $f \in F$ that join along edges $e \in E$. By $\textsc{Del}(\mathbf{P}) := (F_{\textsc{Del}}, E_{\textsc{Del}})$ we refer to a Delaunay triangulation of $\mathbf{P}$, composed of triangular faces $f \in F_{\textsc{Del}} \subset \mathbf{P}^3$ and edges $e \in E_{\textsc{Del}} \subset \mathbf{P}^2$. We will use f and e to both denote the set of vertices from $\mathbf{P}$ and their geometric realizations. Two faces f_1 and f_2 with $f_1 \cap f_2 = e \in E_{\textsc{Del}}$ are called neighbors with common edge e. We consider angles greater or equal $\pi/2$ as obtuse and a triangle with an obtuse interior angle is called obtuse. The longest edge of an obtuse triangle f is opposite to its obtuse angle and will be denoted by $e_f^> = \arg\max_{e \in f} |e|$. Whenever the term $e_f^>$ is used, f implicitly represents an obtuse triangle. For a triangle $f \in F_{\textsc{Del}}$ we denote by $\mathrm{cc}(f)$ its circumcenter. Note that a triangle f is obtuse if it does not contain $\mathrm{cc}(f)$ in its interior and that $\mathrm{cc}(f)$ lies on the opposite side of $e^>$ as the obtuse angle.

We denote by $G = (\mathbf{P}, E)$, a graph over $\mathbf{P}$, where $\mathbf{P}$ represents the set of graph vertices and $E \subset \mathbf{P}^2$ the set of of edges. A sequence of vertices $\rho = (\mathbf{p}_1, \ldots, \mathbf{p}_n)$, $(\mathbf{p}_i, \mathbf{p}_{i+1}) \in E$, is a *path* of length $|\rho| = n$ if none of its edges appears more than once, i.e., edges are mutually distinct. A path is called *closed* if $(\mathbf{p}_n, \mathbf{p}_1) \in E$. We assume a path to be free of loops of length one, i.e., edges $(\mathbf{p}_i, \mathbf{p}_i)$. The Delaunay triangulation itself represents an undirected, connected graph $G_{\textsc{Del}} = (\mathbf{P}, E_{\textsc{Del}})$.

The Voronoi diagram of $\mathbf{P}$ is the partition of $\mathbb{R}^2$ into convex polygons $\mathscr{T}_{\mathbf{p}}$, called *Voronoi tiles* of $\mathbf{p}$, such that every point in $\mathscr{T}_{\mathbf{p}}$ is not farther from $\mathbf{p}$ than from any other vertex in $\mathbf{P}$. For each vertex $\mathbf{p} \in \mathbf{P} \setminus \partial\mathscr{C}(\mathbf{P})$, its Voronoi tile is equal to $\mathscr{T}_{\mathbf{p}} = \mathscr{C}(\{\mathrm{cc}(f) \mid f \in F_{\textsc{Del}}, \mathbf{p} \in f\})$, and we also call $\mathbf{p}$ a Voronoi site.

With these definitions the *Adaptive Delaunay Tessellation* is defined as in [7]:

Definition 1 (Adaptive Delaunay Tessellation). *Let* $\mathbf{P} \subset \mathbb{R}^2$, $\textsc{Del}(\mathbf{P}) = (F_{\textsc{Del}}, E_{\textsc{Del}})$ *and*

$$E^> = \{e_f^> \mid f \in F_{\textsc{Del}} \wedge e_f^> \not\subset \partial\mathscr{C}(\mathbf{P})\}.$$

The tessellation of $\mathscr{C}(\mathbf{P})$ *represented by* $(F_{\textsc{Adt}}, E_{\textsc{Del}} \setminus E^>)$*, where* $F_{\textsc{Adt}}$ *is the set of faces generated by merging triangles with common edges in* $E^>$*, is called the* Adaptive Delaunay Tessellation $\textsc{Adt}(\mathbf{P})$.

Thus, the $\textsc{Adt}$ of a point set is the result of removing from a Delaunay triangulation of $\mathbf{P}$ of each obtuse triangle the longest edge, if this is not a boundary edge. Since no new edges are generated in $\textsc{Adt}(\mathbf{P})$, each triangle $f \in F_{\textsc{Del}}$ is part of some polygon $g \in F_{\textsc{Adt}}$, which we denote by $\mathscr{A}(f) = g \supset f \in F_{\textsc{Del}}$.

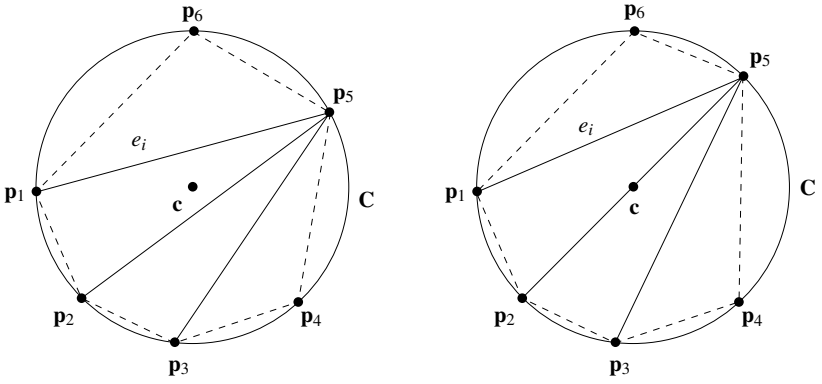


Fig. 2. Points $\mathbf{p}_1, \ldots, \mathbf{p}_n$ on a common circumcircle $\mathbf{C}$ with center $\mathbf{c}$ inside a triangle (a) or on an edge e_j (b).

3 Geometric Properties of the ADT

In this section we will discuss some of the geometric properties of $\text{ADT}(\mathbf{P})$, i.e., uniqueness, connectedness, and the inclusion of Voronoi tiles. The last property states that the Voronoi tile of each $\mathbf{p} \in \mathbf{P}$ is contained in the union of $\mathbf{p}$'s adjacent faces from $\text{ADT}(\mathbf{P})$.

3.1 Uniqueness of the ADT

Proposition 2 (Uniqueness). *For any non-collinear set of points* $\mathbf{P} \subset \mathbb{R}^2$, $\text{ADT}(\mathbf{P})$ *exists and is unique.*

Proof. We will show that $E^>$ contains all edges of E_{DEL} that are non-unique.

For any non-collinear point set $\mathbf{P}$, $\text{DEL}(\mathbf{P})$ exists, and is non-unique, if there are at least $n \geq 4$ points $\mathbf{p}_1, \ldots, \mathbf{p}_n$ with a common empty circumcircle $\mathbf{C}$ with circumcenter $\mathbf{c}$, i.e., $\mathbf{P} \cap]\mathbf{C}[= \emptyset$. The convex hull of these points can be triangulated arbitrarily by edges $e_1, \ldots, e_{n-3}$ without violating the Delaunay condition, see Figure 2(a). Every e_i connects two points not neighboring on the circumcircle and belongs to two triangles of the triangulation of $\mathscr{C}(\{\mathbf{p}_1, \ldots, \mathbf{p}_n\})$. Because there is at most one non-obtuse triangle containing the circumcenter $\mathbf{c}$, every edge e_i belong to at least one obtuse triangle. Furthermore, every edge e_i is the longest edge of the obtuse triangle that lies on the opposite side of e_i with respect to $\mathbf{c}$. Thus, $e_i \in E^>$.

In case $\mathbf{c}$ lies on an edge, say e_j, then $e_j \in E^>$, because at least one of its triangles is rectangular, see Figure 2(b). So, all e_i are in $E^>$, such that $E_{\text{DEL}} \setminus E^>$ is unique. □

Remark 3. If in the above setting the common edge e of two triangles is passing through the common circumcenter, both triangles are rectangular, i.e., obtuse, and e is removed.

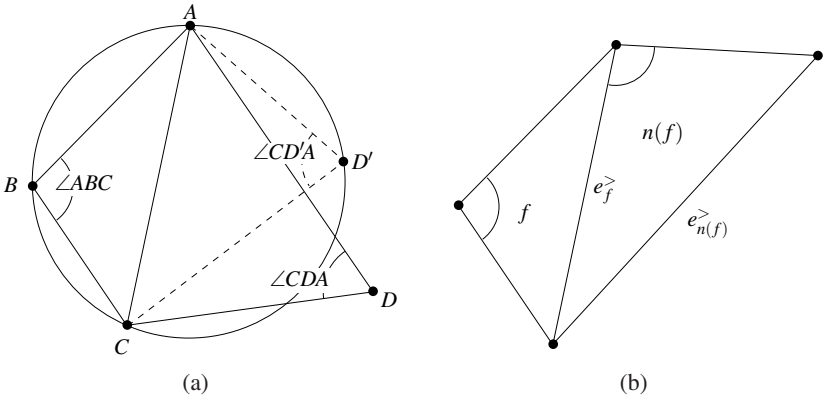


Fig. 3. (a) Two Delaunay triangles with the circumcircle of $\triangle ABC$. (b) $|e_f^>| < |e_{N(f)}^>|$.

3.2 Connectedness of the ADT

Before stating the second geometric property, we introduce some results used in its proof. For every obtuse triangle f with longest edge $e_f^>$ and neighbor f', i.e., $f \cap f' = e_f^>$, we use

$$N(f) := f'$$

to denote neighbor f' of f opposite to the obtuse angle. This relation imposes a subgraph of the dual Delaunay triangulation on the faces of $\text{DEL}(\mathbf{P})$. For non-obtuse triangles or if $e_f^> \subset \partial\mathscr{C}(\mathbf{P})$, we set $N(f) = \emptyset$, and $N(\emptyset) = \emptyset$. This induces the directed graph

$$\begin{aligned} \overrightarrow{G}_{\text{DEL}} &:= (F_{\text{DEL}}, \overrightarrow{E}_{\text{DEL}}), \\ \overrightarrow{E}_{\text{DEL}} &:= \{\, (f, N(f)) \mid f \in F_{\text{DEL}} \wedge N(f) \neq \emptyset \}. \end{aligned} \tag{1}$$

Each $\overrightarrow{e} \in \overrightarrow{E}_{\text{DEL}}$ is associated with one Delaunay edge $e \in E^>$.

In the proof we use the following lemma, which implies that the lengths of the longest edges of obtuse triangles grow along a path in $\overrightarrow{G}_{\text{DEL}}$.

Lemma 4. *For $f \in F_{\text{DEL}}$ with $N(f) \neq \emptyset$,*

$$e_f^> \neq e_{N(f)}^> \Rightarrow |e_f^>| < |e_{N(f)}^>|. \tag{2}$$

Proof. We first show that there cannot be obtuse angles facing the same edge unless they are both right angles. Consider Figure 3(a) and assume $\triangle ABC$ and $\triangle CDA$ with common edge $\overline{AC}$ are triangles in a Delaunay triangulation. Let D' be a point on the circumcircle $\mathbf{C}$ of $\triangle ABC$ right of $\overline{AC}$. Since $\square ABCD'$ is a circular, simple quadrilateral, $\angle ABC + \angle CD'A = \pi$. By the Delaunay criterion, D must either be on or outside $\mathbf{C}$, and comparison to the inscribed angle yields $\angle CDA \leq \angle CD'A = \pi - \angle ABC$. The only obtuse angles that yield equality are $\angle CDA = \angle ABC = \pi/2$, in which case $e_f^> = e_{N(f)}^>$.

In all other cases, i.e., $e_f^> \neq e_{N(f)}^>$, we find $e_f^>$ adjacent to the obtuse angle in $N(f)$ and therefore shorter than $e_{N(f)}^>$, see Figure 3(b). This yields (2). □

Now we can state the result which guarantees that the transformation from Delaunay to adaptive Delaunay does not lead to unconnected elements such as orphaned vertices or even disconnected sub-graphs.

Proposition 5 (Connectedness). *For* $\mathrm{ADT}(\mathbf{P}) = (F_{\mathrm{ADT}}, E_{\mathrm{ADT}})$ *the graph* $G_{\mathrm{ADT}} = (\mathbf{P}, E_{\mathrm{ADT}})$ *is connected.*

Proof. G_{ADT} is a sub-graph of G_{DEL}, and G_{DEL} is connected. An edge of G_{DEL} is not in G_{ADT} if and only if its dual edge is in $\overrightarrow{G}_{\mathrm{DEL}}$. The smallest subset of G_{ADT} which can be disconnected consists of a single vertex. Boundary vertices cannot be disconnected, since the duals of boundary edges are by definition not in $\overrightarrow{G}_{\mathrm{DEL}}$. An inner vertex in a Delaunay triangulation has a minimum valence of three, which means it cannot be disconnected in G_{ADT} unless there is a closed path of length greater or equal to three in $\overrightarrow{G}_{\mathrm{DEL}}$.

We show that a path ρ can only be closed if it has length $|\rho| = 2$. Let without loss of generality $\rho = (f_1, \ldots, f_n)$ be a path in $\overrightarrow{G}_{\mathrm{DEL}}$, i.e., $f_{i+1} = N(f_i)$. First assume $e_{f_i}^> \neq e_{f_{i+1}}^>$, $1 \leq i < n$. With (2) we get

$$|e_{f_1}^>| < |e_{f_2}^>| < \cdots < |e_{f_n}^>|.$$

If ρ were closed, $N(f_n) = f_1$ and $e_{f_n}^> = f_1 \cap f_n$ is both an edge of f_1 and larger than $e_{f_1}^>$. Since this is a contradiction to $e_{f_1}^>$ being the largest edge in f_1, ρ cannot be closed.

Now assume there is an i such that $e_{f_i}^> = e_{f_{i+1}}^>$. Since elements in ρ are mutually distinct and $N(f_{i+1}) = f_i$, f_{i+1} is the last element in ρ and $n = i+1$. For ρ to be closed, $N(f_n) = f_1$ must hold. Thus

$$N(f_n) = N(f_{i+1}) = f_i = f_1,$$

which leaves $i = 1$ and $n = 2$, and ρ can only be closed if it has length 2. Since the longest closed path in $\overrightarrow{G}_{\mathrm{DEL}}$ has length $|\rho| = 2$, G_{ADT} is connected. □

Remark 6 (Implementation). From the above reasoning it appears that numerical instabilities might flip the inequalities and result in ambiguous or even inconsistent results. However, the only setting where the inequalities in (2) are "only just" fulfilled arises with co-circular points, where the uniqueness of the ADT should be forced by allowing an epsilon threshold.

3.3 An Alternative Characterization for the ADT

In the following, we make use of an alternative characterization of the ADT, involving the following construction, see Figures 4(a) and 4(b).

Definition 7 (Polygonal Extension $PE(f)$**).** *For each triangular face* f*, we define*

$$PE(f) =]\mathscr{C}(f \cup \{\mathrm{cc}(f)\})[\cup \{\mathrm{cc}(f)\}$$

as the Polygonal Extension *of* f*, where* $\mathrm{cc}(f)$ *denotes the circumcenter of* f*.*

Note that $PE(f)$ does not contain its boundary except for the circumcenter itself, which is necessary to cover the case of rectangular triangles, as we will see later. We know that if f is non-obtuse, we get

$$\mathrm{cc}(f) \in]f[, \qquad \text{and} \qquad]f[= PE(f). \tag{3}$$

If f is obtuse, then $PE(f)$ extends over $e_f^>$, and

$$\mathrm{cc}(f) \notin]f[, \qquad \text{and} \qquad e_f^> \cap PE(f) \neq \emptyset. \tag{4}$$

The following lemma is required in the proof of Proposition 9.

Lemma 8. *Let f be an obtuse triangle and $f' = N(f) \neq \emptyset$. Then,*

$$PE(f) \setminus f \subset PE(f'). \tag{5}$$

Proof. Consider triangles $f = \triangle ABC$, $f' = \triangle ACD$ and their circumcenters $\mathrm{cc}(f)$, $\mathrm{cc}(f')$ in Figure 4(c). It is sufficient to show that the excess area $PE(f) \setminus f$ given by $]\mathscr{C}(A,C,\mathrm{cc}(f))[$ is already contained in a subset of $PE(f')$ given by $]\mathscr{C}(A,C,\mathrm{cc}(f'))[$, as illustrated in Figure 4(d).

By definition, f shares the $e_f^> = \overline{AC}$ with its neighbor $f' = N(f)$. The circumcenters $\mathrm{cc}(f)$ and $\mathrm{cc}(f')$ lie on the perpendicular bisector of $e_f^>$ on the same side of $e_f^>$ as D. By the Delaunay empty circumcircle criterion, D is outside the circumcircle of f, and on the side of $e_f^>$ opposite to B, thus the distance of $\mathrm{cc}(f')$ to $e_f^>$ is greater or equal to that of $\mathrm{cc}(f)$. Consequently,

$$\mathscr{C}(\{A,C,\mathrm{cc}(f)\} \subset \mathscr{C}(\{A,C,\mathrm{cc}(f')\}) \subset \mathscr{C}(\{A,C,\mathrm{cc}(f'),D\}).$$

This relation also applies to the interior of these sets, which means

$$\begin{aligned} PE(f) \setminus f &=]\mathscr{C}(\{A,C,\mathrm{cc}(f)\})[\\ &\subset]\mathscr{C}(\{A,C,\mathrm{cc}(f'),D\})[\cup \{\mathrm{cc}(f')\} = PE(f'), \end{aligned}$$

see Figure 4(d). Note that in case of a rectangular triangle, the circumcenter lies on its obtuse edge, and $]\mathscr{C}(\{A,C,\mathrm{cc}(f)\})[= \emptyset \subset PE(f')$ for arbitrary f'. □

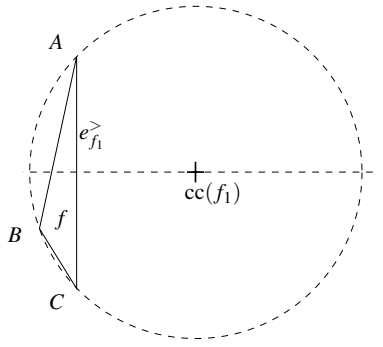
Lemma 8 implies an equivalent characterization for the set of edges $E^>$ that are removed when transforming a Delaunay triangulation into an ADT.

Proposition 9. *Let* $\mathrm{DEL}(\mathbf{P}) = (F_{\mathrm{DEL}}, E_{\mathrm{DEL}})$ *and* $E^>$ *as in Definition 1. Then*

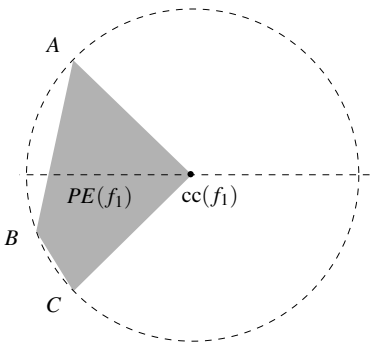
$$e \in E^> \quad \Leftrightarrow \quad \exists f \in F_{\mathrm{DEL}} : \; e \cap PE(f) \cap]\mathscr{C}(\mathbf{P})[\neq \emptyset.$$

So, an adaptive Delaunay tessellation can also be created by merging all triangles, whose polygonal extensions intersect, into polygons.

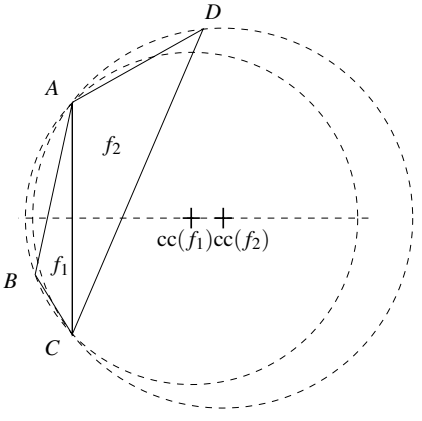
Proof. "⇒": If $e \in E^>$, then $e = e_f^>$ for some obtuse $f \in F_{\mathrm{DEL}}$, and $e \cap]\mathscr{C}(\mathbf{P})[\neq \emptyset$. Then, (4) yields also $e \cap PE(f) \cap]\mathscr{C}(\mathbf{P})[\neq \emptyset$.



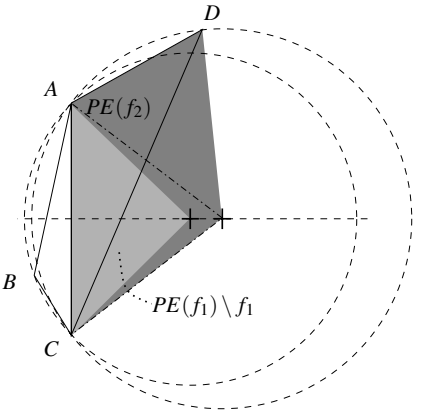
(a) Triangle f and its circumcenter $\mathrm{cc}(f)$.



(b) Polygonal extension $PE(f)$.



(c) Triangles f and f'.



(d) $PE(f) \setminus f$ in light gray and $PE(f')$.

Fig. 4. Geometric interpretation of the polygonal extension $PE(f)$

"$\Leftarrow$": Let $e \cap PE(f) \cap]\mathscr{C}(\mathbf{P})[\neq \emptyset$ for an $f \in F_{\mathrm{DEL}}$. If e is an edge of f then $e = e^{>}_{f}$ and $e \in E^{>}$. Otherwise, $e \neq e^{>}_{f}$, and $e \cap PE(f) \setminus f \neq \emptyset$. By Lemma 8 there exists an obtuse triangle f' such that $e \cap PE(f') \neq \emptyset$. This argument applies iteratively until e is an edge of f'. Since $|F_{\mathrm{DEL}}|$ is finite and the subset relation in Lemma 8 is strict, this iteration terminates with a $f' \in F_{\mathrm{DEL}}$ such that e is an edge of f'. The final f' exists, otherwise e would intersect the interior of a triangle and $\mathrm{DEL}(\mathbf{P})$ was no triangulation. □

The relation of triangles, polygonal extensions and ADT faces is captured in the following result.

Proposition 10. *For every $f \in F_{\mathrm{DEL}}$,*

$$f \quad \subseteq \quad [PE(f)] \cap \mathscr{C}(\mathbf{P}) \quad \subseteq \quad \mathscr{A}(f). \tag{6}$$

Proof. The left inclusion is true by construction of $PE(f)$ and $f \subseteq \mathscr{C}(\mathbf{P})$.

The right inclusion is more difficult to show. If a triangle f is non-obtuse, or rectangular, $\mathrm{cc}(f) \in f$ and $[PE(f)] = f \subseteq \mathscr{A}(f)$. In all other cases, f is obtuse and $\mathrm{cc}(f) \notin f$. If $e_f^{>}$ is on the boundary of $\mathscr{C}(\mathbf{P})$, then $[PE(f)] \cap \mathscr{C}(\mathbf{P}) = f \subseteq \mathscr{A}(f)$. Otherwise, f is obtuse and there is a triangle $f' = N(f)$ such that

$$
\begin{aligned}
& PE(f) \setminus f \subseteq PE(f') && \text{by (5)},\\
\Rightarrow\quad & PE(f) \subseteq f \cup PE(f'),\\
\Rightarrow\quad & [PE(f)] \subseteq f \cup [PE(f')],\\
\Rightarrow\quad & [PE(f)] \cap \mathscr{C}(\mathbf{P}) \subseteq f \cup [PE(f')] \cap \mathscr{C}(\mathbf{P}).
\end{aligned}
$$

Since $\mathscr{A}(f) = \mathscr{A}(f')$, the claim holds if we can prove $[PE(f')] \cap \mathscr{C}(\mathbf{P}) \subseteq \mathscr{A}(f')$. The argument applies repeatedly until f' is non-obtuse, rectangular, or has its obtuse edge on the boundary of $\mathscr{C}(\mathbf{P})$ following a path ρ in $\overrightarrow{G}_{\mathrm{DEL}}$. In the proof for Proposition 5 we showed that every path ρ ends in a triangle f that is either non-obtuse, has $e_f^{>}$ on the convex hull, or in a loop consisting of two rectangular triangles. Therefore, the repeated application of the argument terminates and the claim is proved. □

An important observation following immediately from the above Proposition is given in the following corollary:

Corollary 11. *For every* $f \in F_{\mathrm{DEL}}$ *with* $\mathrm{cc}(f) \in \mathscr{C}(\mathbf{P})$: $\mathrm{cc}(f) \in \mathscr{A}(f)$.

3.4 Coverage of Voronoi Tiles

The last property we are going to prove is the coverage of a vertex' Voronoi tile by its adjacent ADT tiles, as illustrated in Figure 5(a). With this property, the adjacency information contained in the ADT is sufficient to access the whole area covered by the Voronoi cell of a vertex.

Proposition 12 (Inclusion of Voronoi Tile). *If* $\mathscr{T}_{\mathbf{p}}$ *denotes the Voronoi tile of* $\mathbf{p} \in \mathbf{P}$*, then*

$$\mathscr{T}_{\mathbf{p}} \cap \mathscr{C}(\mathbf{P}) \subset \bigcup_{\substack{g \in F_{\mathrm{ADT}} \\ g \ni \mathbf{p}}} g.$$

Proof. First assume that $\mathbf{p} \notin \partial\mathscr{C}(\mathbf{P})$, i.e., $\mathscr{T}_{\mathbf{p}}$ is finite as in Figure 5(a), $\mathbf{p}$ has a set of adjacent triangles $\{f\}_{f \ni \mathbf{p}}$, which in turn are adjacent via the edges $\{e\}_{e \ni \mathbf{p}}$. For each edge $e \ni \mathbf{p}$ and triangles $f_1 \cap f_2 = e$, we get a new triangle $A_e = \mathscr{C}(\{\mathbf{p}, \mathrm{cc}(f_1), \mathrm{cc}(f_2)\})$, see Figure 5(b). This partitions $\mathscr{T}_{\mathbf{p}}$,

$$\mathscr{T}_{\mathbf{p}} = \bigcup_{e \ni \mathbf{p}} A_e.$$

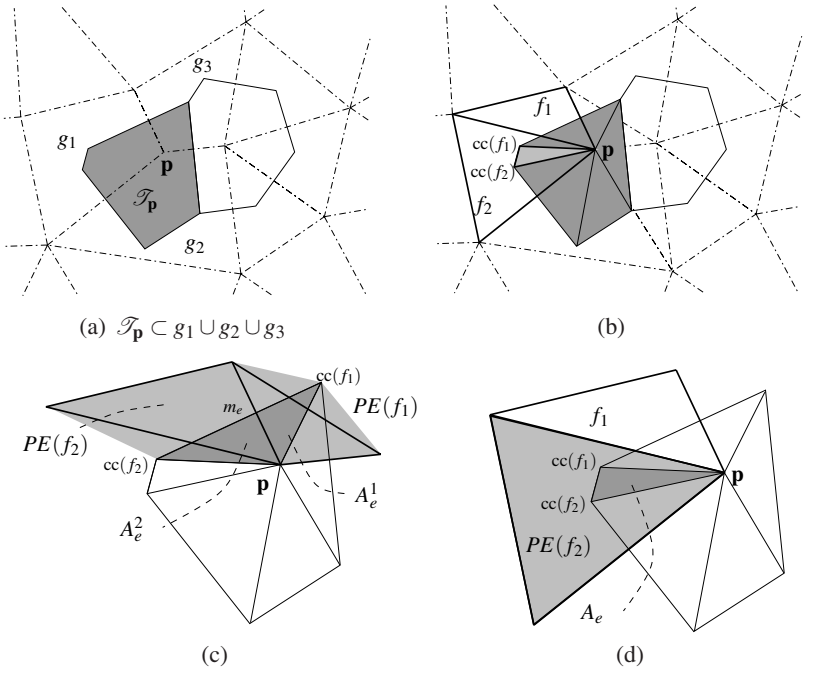


Fig. 5. Voronoi coverage of adjacent ADT tiles

So we will show that each $A_e \cap \mathscr{C}(\mathbf{P})$ is covered by $\mathscr{A}(f_1) \cup \mathscr{A}(f_2)$ for two cases:

- $\mathrm{cc}(f_1)$ and $\mathrm{cc}(f_2)$ are on opposite sides of e, see Figure 5(c): So $\mathrm{cc}(f_1)$ and $\mathrm{cc}(f_2)$ are located on the perpendicular bisector of e that intersects e in m_e. We can split A_e into $A_e^1 = \mathscr{C}(\{\mathbf{p}, \mathrm{cc}(f_1), m_e\})$ and $A_e^2 = \mathscr{C}(\{\mathbf{p}, \mathrm{cc}(f_2), m_e\})$ with $A_e = A_e^1 \cup A_e^2$. Since $m_e, \mathbf{p}, \mathrm{cc}(f_1) \in [PE(f_1)]$, (6) guarantees that $A_e^1 \cap \mathscr{C}(\mathbf{P}) \subset \mathscr{A}(f_1)$ and analogously for A_e^2.
- $\mathrm{cc}(f_1)$ and $\mathrm{cc}(f_2)$ are on the same side of or on e, see Figure 5(d): Without loss of generality, let f_1 be obtuse. We know $\mathrm{cc}(f_1) \in PE(f_1) \setminus f_1$ and by (5), $\mathrm{cc}(f_1) \in PE(f_2)$. Since $\mathrm{cc}(f_2) \in PE(f_2)$ as well, $\mathrm{cc}(f_1), \mathrm{cc}(f_2), \mathbf{p} \in [PE(f_2)]$, and from the convexity of $[PE(f_2)]$ it follows that $A_e \cap \mathscr{C}(\mathbf{P}) \subset PE(f_2) \cap \mathscr{C}(\mathbf{P}) \subset \mathscr{A}(f_2)$.

Now consider the case that $\mathbf{p} \in \partial\mathscr{C}(\mathbf{P})$, and $\mathscr{T}_\mathbf{p}$ becomes infinite. The argument goes as above except for the two edges $e_1, e_2 \subset \partial\mathscr{C}(\mathbf{P})$ adjacent to $\mathbf{p}$ which have only one adjacent triangle each, say f_1 respective f_2, as illustrated in Figure 6(a). Corresponding to the above two cases, let without loss of generality $\mathrm{cc}(f_1) \notin \mathscr{C}(\mathbf{P})$ like in Figure 6(b). Because A_{e_1} is outside the convex hull, it does not contribute to $\mathscr{T}_\mathbf{p} \cap \mathscr{C}(\mathbf{P})$ and needs not be considered. Now, let without loss of generality $\mathrm{cc}(f_2) \in \mathscr{C}(\mathbf{P})$ like in Figure 6(c). The contribution of A_{e_2} to $\mathscr{T}_\mathbf{p} \cap \mathscr{C}(\mathbf{P})$ is given by $\mathscr{C}(\mathbf{p}, m_{e_2}, \mathrm{cc}(f_2)) \subset f_2 \subset \mathscr{A}(f_2)$.

Consequently, all parts $A_e \cap \mathscr{C}(\mathbf{P})$ are covered by ADT polygons, which are the covers of adjacent Delaunay triangles and thus also adjacent to $\mathbf{p}$. □

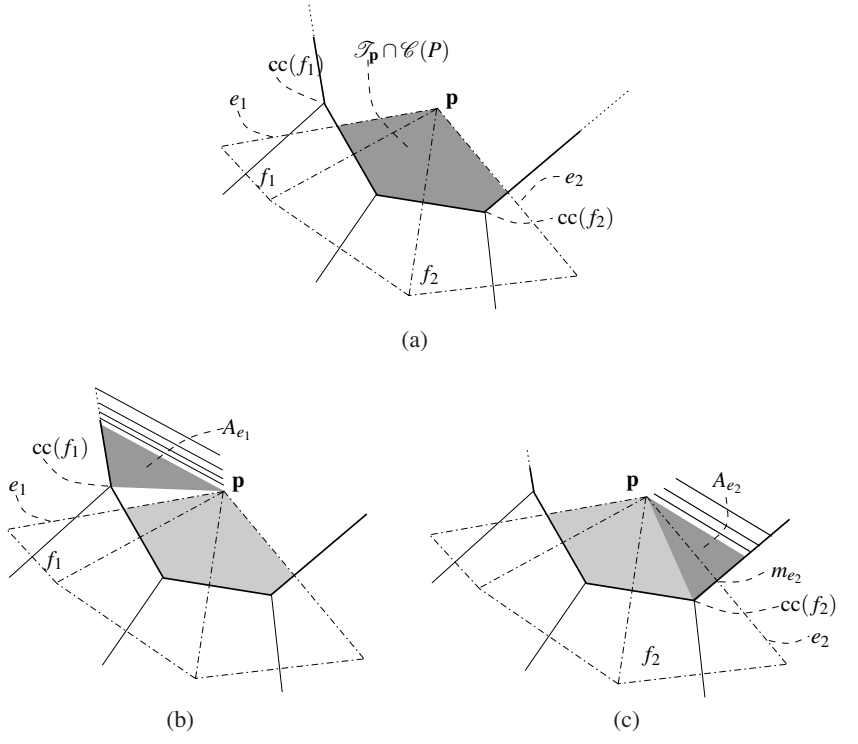


Fig. 6. Coverage of ADT tiles on the boundary of $\mathscr{C}(\mathbf{P})$. (a) For $\mathbf{p} \in \partial\mathscr{C}(\mathbf{P})$, e_1 and e_2 have only one adjacent triangle each. (b) Contribution of A_{e_1} to $\mathscr{T}_\mathbf{p}$. (c) Contribution of A_{e_2} to $\mathscr{T}_\mathbf{p}$.

4 Discussion

The ADT is optimal only under specific conditions, i.e., if the vertex positions cannot be changed, and the Voronoi tile of each vertex needs to be covered by adjacent polygons. A very homogeneous point set with all non-obtuse triangles is shown in Figure 7(a). In this setting, the Delaunay triangulation and the ADT are identical. Figure 7(b) shows a point set in which ill-shaped triangles are merged into ADT polygons, thus improving the overall tile shape. Figure 7(c) demonstrates a situation where the ADT produces non-convex polygons, and the extreme case of a "hanging edge" is illustrated in Figure 7(d). Apart from these artifacts inside the convex hull, which are rarely faced in practice, badly shaped polygons frequently occur near the convex hull as a result of many degenerate triangles, as illustrated in Figure 7(e). Arguably, the boundary polygons are of better shape than the triangles they are composed of. In [7], a novel polygon condition number has been introduced as a quality measure of polygonal tessellations. Extensive statistical analysis of the ADT has shown that it considerably improves the mean condition number of most unstructured triangulations over interior elements, and results in moderate improvement for boundary elements. The issues inside the convex hull can be handled by allowing vertices to move, and a regularization step could lead to

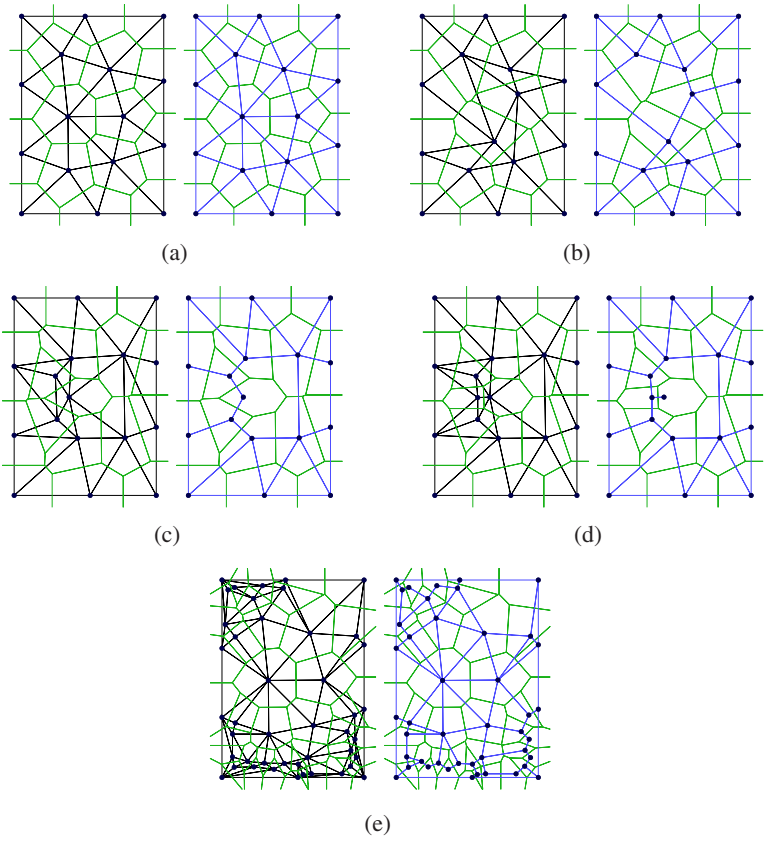


Fig. 7. Several levels of badness in the point cloud. (a) very regular, (b) some obtuse triangles, (c) constellation leading to non-convex ADT tiles, (d) constellation leading to flapping edge, (e) bad boundary elements.

more well-shaped triangles and thus better ADT-polygons. Furthermore, the boundary issues can be solved by insertion of an adequate amount of vertices along the convex hull.

5 Conclusion

Certain nodal integration schemes, which are meshless extension of the Finite Element Method, lead to new requirements on the background tessellation of the domain which are no longer met by the popular Delaunay triangulation. Particularly nodal integration schemes based on the Voronoi diagram of the domain vertices entail the integration of an interpolant along the boundary of Voronoi tiles. This leads to a desirable property of the background tessellation referred to as the *coverage of Voronoi tiles*. To this end, the Adaptive Delaunay Tessellation for planar domains has been introduced before, but no rigorous proofs have been given.

In this paper we have introduced a simple, constructive definition of the Adaptive Delaunay Tessellation based on the Delaunay triangulation of a point set, and showed that it is unique, connected, and has the coverage of Voronoi tiles property. We furthermore provided a critical assessment of some shortcomings of the ADT that need to be considered in applications. This provides a profound theoretical background for the application of this new tessellation for nodal integration schemes in the Finite Element Method.

Acknowledgements

This work was supported by the international graduate school DFG grant 1131 on "Visualization of Large and Unstructured Data Sets - Applications in Geospatial Planning, Modeling and Engineering".

References

1. Shewchuk, J.: What is a good linear element? Interpolation, conditioning, and quality measures. In: Proc. Eleventh Int. Meshing Roundtable, Ithaca, New York, pp. 115–126. Sandia National Laboratories (2002)
2. Lohner, R.: Progress in grid generation via the advancing front technique. Eng. with Computers 12, 186–210 (1996)
3. Tournois, J., Alliez, P., Devillers, O.: Interleaving delaunay refinement and optimization for 2D triangle mesh generation. In: Proc. 16th Int. Meshing Roundtable (2007)
4. Bossen, F., Heckbert, P.: A pliant method for anisotropic mesh generation. In: Proc. 5th Int. Meshing Roundtable, pp. 63–74 (1996)
5. Antani, L., Delage, C., Alliez, P.: Mesh sizing with additively weighted voronoi diagrams. In: Proc. 16th Int. Meshing Roundtable (2007)
6. Hjelle, Ø., Dæhlen, M.: Triangulations and Applications. Mathematics and Visualization. Springer, Secaucus (2006)
7. Constantiniu, A., Steinmann, P., Bobach, T., Farin, G., Umlauf, G.: The adaptive delaunay tessellation: A neighborhood covering meshing technique. Computational Mechanics 42(5), 655–669 (2008)
8. Sukumar, N., Moran, B., Semenov, A., Belikov, V.: Natural neighbour Galerkin methods. Int. J. for Numerical Methods in Engineering 50, 1–27 (2001)
9. Chen, J., Wu, C., Yoon, S., You, Y.: A stabilized conforming nodal integration for Galerkin mesh-free methods. Int. J. Numer. Methods Eng. 50, 435–466 (2001)

Quadrangular Parameterization for Reverse Engineering

David Bommes, Tobias Vossemer, and Leif Kobbelt

RWTH Aachen University

Abstract. The aim of Reverse Engineering is to convert an unstructured representation of a geometric object, emerging e.g. from laser scanners, into a natural, structured representation in the spirit of CAD models, which is suitable for numerical computations. Therefore we present a user-controlled, as isometric as possible parameterization technique which is able to prescribe geometric features of the input and produces high-quality quadmeshes with low distortion. Starting with a coarse, user-prescribed layout this is achieved by using affine functions for the transition between non-orthogonal quadrangular charts of a global parameterization. The shape of each chart is optimized non-linearly for isometry of the underlying parameterization to produce meshes with low edge-length distortion. To provide full control over the meshing alignment the user can additionally tag an arbitrary subset of the layout edges which are guaranteed to be represented by enforcing them to lie on iso-lines of the parameterization but still allowing the global parameterization to relax in the direction of the iso-lines.

Keywords: reverse engineering, quadrangular remeshing, global parameterization.

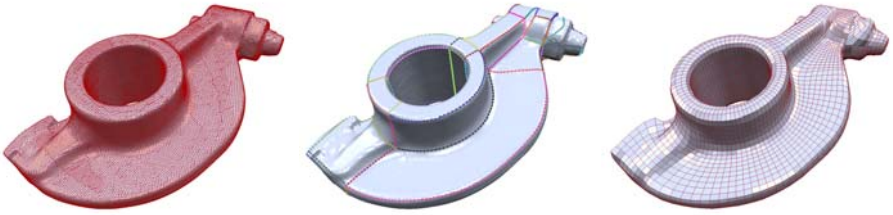
1 Introduction

The quality of a quadrangulation depends on several, often conflicting, properties. The most important ones are *regularity, alignment, orientation* and *sizing*. *Regularity* means that the number of irregular vertices (valence $\neq 4$) should be minimized, while *orientation* implies that the elements should be locally oriented to the principal curvature directions to best capture the geometry. For geometries with sharp features *alignment* plays an important role, i.e. feature lines should be explicitly represented through edges in the quadrangulation, which is an even stronger requirement than *orientation*. Finally *sizing* means that the edge-length should be nearly constant which can be interpreted as providing a close to isometric inherent parameterization in regular regions of the mesh.

Quadrangulations optimizing these properties are perfectly suited for many computer graphics applications like Modeling, e.g. as structure aligned control meshes for Catmull-Clark subdivision, multi-level hierarchies or shape matching to name a few.

The aim of Reverse Engineering is to convert a given unstructured triangle mesh into such a structured quadrangulation (see Figure 1). Full automatic generation has proven to be a very hard global task and often design decisions depend on the intended usage and cannot forecast by only looking at the geometry. Therefore in Reverse Engineering full user-control is desired where the user can easily provide the topology, i.e. the

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 55–69, 2010.



(a) (b) (c)

Fig. 1. Reverse Engineering Pipeline: (a) The input is a dense, unstructured triangle mesh. (b) The user provides a coarse layout controlling the quadrangulation. Singularities can only occur at nodes of this layout. (c) A distortion minimizing parameterization is computed to extract a pure quadmesh.

number and position of singularities, and some alignment constraints for the resulting mesh. This can be achieved in a simple way by using coarse layouts which partition the surface in quadrilateral patches as illustrated in Figure 1 (b). From this layout a globally smooth parameterization can be computed by assigning a two-dimensional chart to each patch and connecting the parameterizations of neighboring charts with so called transition functions (see Figure 1 (c)).

The resulting mesh quality strongly depends on the metric distortion of the parameterization and on the alignment to sharp features. Consequently we present a new method to handle both tasks in a robust way, enabling the usage of global parameterization techniques for Reverse Engineering. Our main contribution is a chart optimization technique which minimizes the distortion of the resulting global parameterization. In contrast to other methods each chart is allowed to be an arbitrary 5 degree-of-freedom (DOF) quadrilateral with interior angles possibly differing from 90 degrees. As a result we need to specify generalized transition functions between these charts. Other important ingredients of our practical Reverse Engineering method are alignment constraints and T-Vertices, enabling simplified layout design. Figure 2 illustrates the gain of quality due to our chart optimization where chart corners are chosen to form a unit square (a), an optimized rectangle (b) and an optimized general quadrilateral (c).

2 Related Work

Remeshing of surfaces was investigated intensively in the last years and there are many good surveys on this topic such as [1] and [2]. Therefore we will only discuss the most relating works.

Direct quadrangulation methods like [3, 4] trace lines of principal curvatures to generate a quaddominant mesh. These methods have no explicit control over regularity and placement of singularities which is necessary for Reverse Engineering.

Global parameterization methods as introduced by Khodakovsky et al. [5] take a different way. In most cases a multi-chart parameterization is computed where continuity conditions between neighboring charts ensure that all mappings of the integer grid

$\mathbb{Z} \times \mathbb{Z}$ stitch together compatibly and form a mesh consisting of quadrilaterals only. Some methods use a guiding vector field, often derived from the principal curvatures of a surface, for local element orientation [6, 7, 8]. However, the creation of suitable guiding fields is alone a very hard problem. Recently Ray et al. proposed a method to design smooth N-symmetric fields with prescribed singularities [9]. But the automatic placement of singularities is still unsolved.

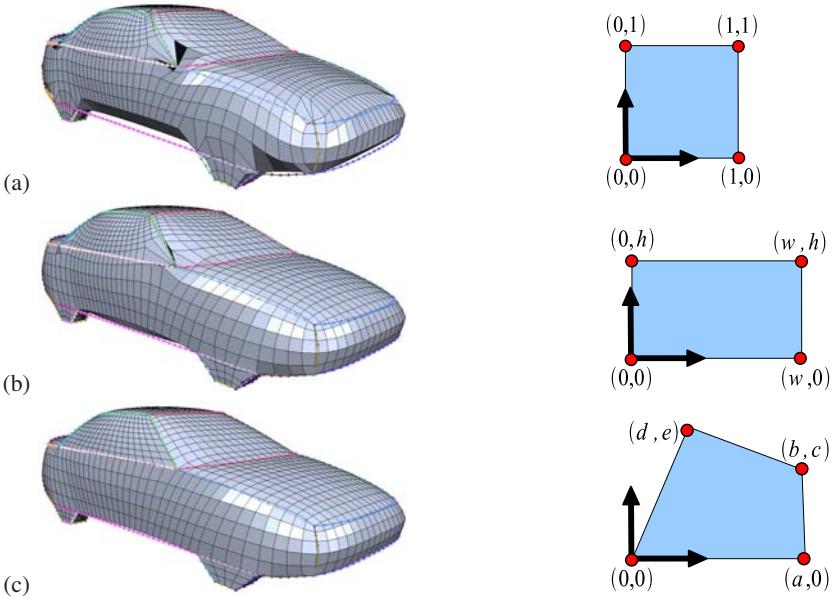


Fig. 2. (a) A global parameterizations using unit charts leads to large distortions and foldovers for a simple car model. (b) Even with optimized rectangular domains the distortions get large where the patches have a trapezoidal shape. (c) Our generalized, quadrangular parameterization leads to low distortion. Notice that in this example no alignment constraints were applied.

Instead of a vector field a coarse layout can be used to define the partitioning of the surface into patches for a multi-chart parameterization and define the rough element orientation. For user-controlled Reverse Engineering this way is very appealing and intuitive. Dong et al. used the Morse-Smale complex of Laplacian eigenfunctions to automatically compute a layout [10]. This method needs only a few singularities but they are placed at almost random positions and there is no element orientation w.r.t. geometric relevant structures. Huang et al. added orientation and alignment control to this method by incorporating a guiding vector field [8].

A user designed coarse layout, here called singularity graph, was also used in [11] to compute globally smooth harmonic parameterizations. These layout-based techniques are closely related to our method. Therefore we will discuss them in more detail.

The method of Dong et al. [10] uses simple unit squares as charts for a globally smooth parameterization. This is justified because in their layout, neighboring surface

patches, originating from the Morse-Smale complex of the Laplacian eigenfunction, have similar size. Furthermore the layout vertices, representing the mesh singularities, are relaxed on the surface to prevent foldovers and large distortions. In Reverse Engineering such a relaxation technique is not reasonable since it interferes with the desired user-control. Figure 2 (a) shows the result of a globally smooth parameterization onto unit square shaped charts with a fixed user provided layout. As one can see there are large distortions and foldovers reflecting the fact that neighboring surface patches are far from being equally sized. So obviously unit square charts are not sufficient for our setting.

If one would restrict the layout to quadrilaterals and choose all free coefficients to one the globally smooth parameterization technique of Tong et al. [11] is exactly the same as the one discussed in the last paragraph. Notice that this equality is non-trivial since both papers use a different formalism to derive the final global linear system. Besides, the parameterization of Tong et al. is more general because it allows a larger class of charts. Each chart is a polygon where the vertices lie on integer positions and all edges are aligned to the coordinate axes, accordingly all interior angles of a chart are multiples of 90 degree. In our car example this means moving from unit squares to rectangular charts with two DOF's, namely the two independent edge lengths. In the original method this new DOF's are chosen manually or by using a heuristic which simply rounds the length of the corresponding layout edges to integer. Figure 2 (b) shows the result of the car example using rectangular charts. Here we already used our chart optimization technique presented in Section 3.1, instead of their heuristic, to minimize the length distortion. But still we observe large distortions, for example near the corner of the front window.

The problem is that the surface patches are far from being rectangular. Consequently we consider an even more general class of charts, i.e. we allow charts to be arbitrary quadrilaterals with five DOF's. We exploit these DOF's to minimize the distortion of the parameterization and the result can be seen in 2 (c). Even without alignment constraints the quadmesh edges follow the user prescribed layout and the length distortions are much lower. This introductory example motivates our design choices for a practical Reverse Engineering method. In the subsequent paragraphs our method will be explained in more detail.

3 Quadrangular Global Parameterization

The input to our quadrangular multi-chart parameterization method is a triangle mesh $M = (V, E, F)$ of arbitrary genus, which is a set of vertices, edges and faces, and a layout graph $G = (\mathscr{V}, \mathscr{E}, \mathscr{F})$. For each edge of the layout graph the user can additionally set a tag which enforces the alignment of the parameterization onto this layout edge, as described in Section 3.2. The scenario of a multi-chart parameterization is depicted in Figure 3. The vertices of the layout graph (red points) lie at triangle mesh vertices and each edge of the layout graph intersects several mesh edges (blue points). In this way all mesh vertices are partitioned into several surface patches, which are disjoint except the layout vertices that belong to all neighboring patches. Each such patch P_α is equipped with a two-dimensional chart C_α. Assume for simplicity that each layout graph face has

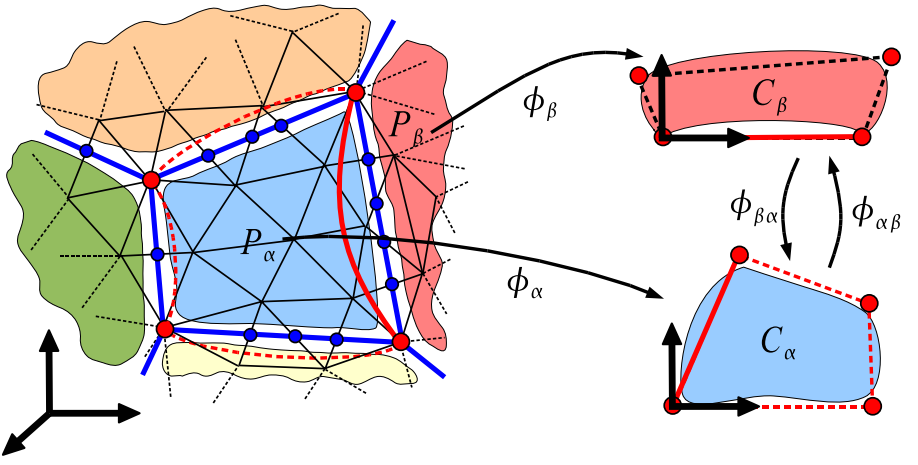


Fig. 3. The left part shows the layout of a multi-chart parameterization. Vertices of the layout graph (red) lie at mesh vertices and edges of the layout graph (blue) cut several mesh edges. Each inner vertex of a patch P_α stores its parameter coordinates w.r.t the local frame of chart C_α. For all pairs of neighboring patches transition functions $\phi_{\alpha\beta}$ exist which translate between their charts. Notice that the red quadrilateral, connecting the four corners of Chart C_α, mapped to the surface is generally not identical to the blue layout.

exactly four vertices, we will discuss in Section 3.3 how to incorporate more general settings. The task is now to compute a piecewise linear multi-chart parameterization, i.e. each vertex $v_i \in \mathbb{R}^3$ belonging to P_α is mapped by the function ϕ_α to the parameter coordinates $u_i^\alpha \in \mathbb{R}^2$ expressed w.r.t the frame of chart C_α. Additionally for triangles with vertices in different patches, for instance P_α and P_β, we need a transition function $\phi_{\alpha\beta}$ to translate between their charts in order to parameterize them. Obviously both directions are possible and inverse to each other $\phi_{\beta\alpha} = \phi_{\alpha\beta}^{-1}$ and the transition from one chart into itself is simply the identity $\phi_{\alpha\alpha} = Id_2$.

A discrete harmonic parameterization of a surface with disc topology mapping to a single chart is a well studied topic where typically the boundary of the surface is mapped to the boundary of a disc and each interior vertex has to fullfill the discrete harmonic equations

$$\sum_{j \in N_i} \bar{w}_{ij}(u_j - u_i) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{1}$$

where N_i are the one-ring neighbors of vertex v_i and $\bar{w}_{ij}$ are normalized edge weights which sum to one $\sum_{j \in N_i} \bar{w}_{ij} = 1$. In all our examples we used the normalized discrete harmonic weights

$$w_{ij} = \frac{1}{2}(cot\,\alpha_{ij} + cot\,\beta_{ij}) \quad with \quad \bar{w_{ij}} = w_{ij} / \sum_{j \in N_i} \bar{w}_{ij} \tag{2}$$

where α_{ij} and β_{ij} are the two angles opposite to edge e_{ij}. There are many other good choices like Floater's Mean Value Coordinates, see [2] for more details. The key observation is that in our multi-chart parameterization setting we can compute a harmonic

parameterization in the same way. The only difference is that instead of fixing a whole boundary we now only fix the corner vertices of the layout graph in each chart and use the transition functions to compute the harmonic conditions in a common frame:

$$\sum_{(j,\beta)\in N_i} \bar{w}_{ij}(\phi_{\beta\alpha}(u_j^\beta) - u_i^\alpha) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{3}$$

In this formulation a global relaxation is achieved. If the transition functions are affine the combination of the above equations for all free vertices is a global linear system of dimension $2(n-k) \times 2(n-k)$ where n is the number of triangle mesh vertices and k is the number of layout vertices. The translational part of the affine transition function as well as known values of constrained layout corners are moved to the right-hand-side. Remember that the coordinates of layout corners cannot be unique because they belong to different charts with different frames. So we need to specify $4|\mathscr{F}|$ many corner positions.

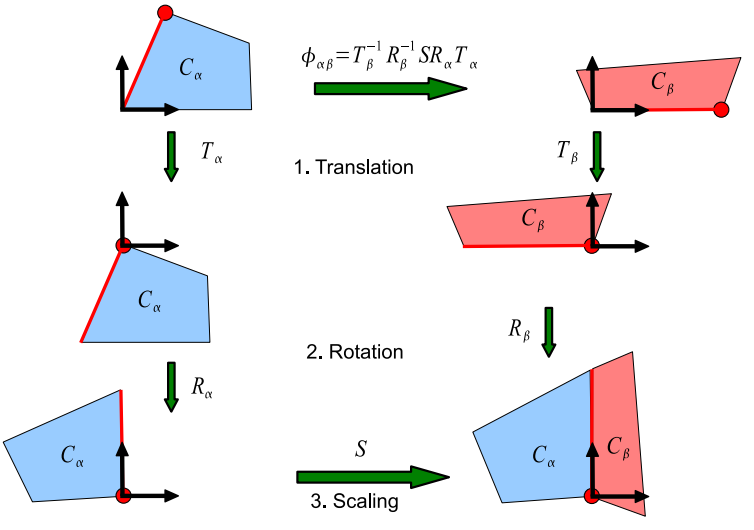


Fig. 4. A common coordinate frame of two charts C_α and C_β can be constructed by first translating a common point into the origin, then rotating the common edge to a coordinate axis and finally scaling along this axis to end up with the same edge length. The transition functions between the charts are constructed in the same way by using the inverse of either α or β functions.

These parameter coordinates of the four patch corners can be in general position (keeping the same orientation as on the surface). However we choose the first one to be the origin and the second one to lie on the first coordinate axis which makes the representation unique. So we end up with five DOF's (a,b,c,d,e) for an arbitrary quadrilateral (see Figure 2 (c)). The transition function between neighboring charts which share a common edge (red) are simple affine functions, combinations of translations, rotations and a scaling as depicted in Figure 4.

$$\phi_{\alpha\beta} = T_\beta^{-1} R_\beta^{-1} S R_\alpha T_\alpha \tag{4}$$

They can be precomputed as 3×3 matrices in extended coordinate representation before accumulating the resulting values into the global system matrix.

The only question left is how to choose adequate corner parameter coordinates (a,b,c,d,e) for a given patch. In [11] the average length of two opposing layout edges rounded to an integer was used to fix width and height of the corresponding rectangle. In the case of a five DOF chart we could do something similar by using all lengths of the patch's boundary. But as explained in the next section the available DOF's can be used to optimize the resulting parameterization in a more founded but still efficient way, which in general leads to better results.

3.1 Chart Optimization

The idea of our chart optimization algorithm is to minimize the metric distortion of the parameterization ϕ. The local distortion near a surface point p_0 in direction v (in local coordinates of the tangent plane) is described by the first order Taylor expansion

$$\phi(p_0+v) \approx \phi(p_0) + J_\phi(p_0)v \quad \Rightarrow \quad \phi(p_0+v) - \phi(p_0) \approx J_\phi(p_0)v \tag{5}$$

where J_ϕ is the Jacobi matrix which can be written as two rotations and a scaling by applying the singular value decomposition

$$J_\phi = U \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} V^T \tag{6}$$

Mapping a unit length vector $\|v\| = 1$, lying in the tangent plane of p_0, into its chart the resulting vector has length $\|J_\phi v\| \in [\sigma_1, \sigma_2]$. Consequently a circle on the surface is mapped to an ellipse in the chart as illustrated in Figure 5. There are some well known special cases [2]:

1. $\sigma_1 = \sigma_2$ is a conformal mapping which maps circles to scaled circles
2. $\sigma_1 \cdot \sigma_2 = 1$ is an equiareal mapping
3. $\sigma_1 = \sigma_2 = 1$ is an isometric mapping with no distortion

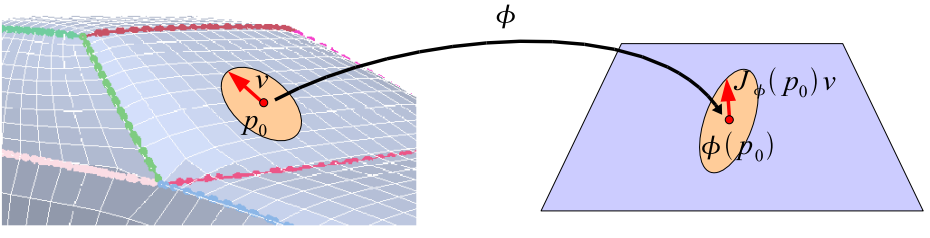


Fig. 5. Mapping a small disc from the tangent plane around a point p_0 the transformation can be approximated by the Jacobi matrix J_ϕ of the mapping ϕ. This means mapping circles into ellipses where the length of the principal axes are related to the singular values of J_ϕ.

Clearly an isometric mapping is the best we can hope for. So we try to choose our chart corners to get as isometric as possible. The desired isometry measure is $E = |\sigma_1 - 1| + |\sigma_2 - 1|$. To approximate this measure we take the quadratic Frobenius norm of the 2D strain tensor

$$E = \|J_\phi^T J_\phi - I\|_2^2 \tag{7}$$

which is 0 in the case of isometry and $(\sigma_1^2 - 1)^2 + (\sigma_2^2 - 1)^2$ when the mapping is conformal.

Using a triangle mesh where the mapping is piecewise-linear, the Jacobi-matrix of a triangle is constant and depends linearly on the parameter values u_0, u_1 and u_2 of the triangle,

$$J_\phi = [u_0 u_1 u_2] \begin{bmatrix} p_0 & p_1 & p_2 \\ 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{8}$$

In the above equation p_0, p_1 and p_2 are the 3D triangle vertices in local 2D coordinates and u_0, u_1 and u_2 are the corresponding parameter values. Therefore $J_\phi^T J_\phi$ is quadratic and the isometry measure of a triangle t is a quartic polynomial in the parameter values, $E_t(u_0, u_1, u_2) = \|J_\phi^T J_\phi - I\|_2^2$.

The aim of this section is to optimize the isometry of the harmonic parameterization by finding adequate parameter coordinates for the four corners of a chart. Consequently we need to express the isometry measure of a triangle w.r.t. these values (a, b, c, d, e). To approximate the relation between the global parameterization and the change of chart corner positions we assume that the dependency is bilinear, which is a good approximation for all interior vertices of a chart:

$$u_i = u_i(a, b, c, d, e) = s_i(1 - t_i) \begin{pmatrix} a \\ 0 \end{pmatrix} + (1 - s_i) t_i \begin{pmatrix} b \\ c \end{pmatrix} + s_i t_i \begin{pmatrix} d \\ e \end{pmatrix}. \tag{9}$$

Since we use these bilinear coordinates s_i and t_i in the sense of freeform deformation, the parameter coordinates u_i are linear in the corner positions (a, b, c, d, e) and so the measure $E_t(a, b, c, d, e)$, now expressed in dependency of the four chart corners, is still a quartic polynomial. Finally we sum up the measures of all triangles lying completely inside the polygon formed by the chart corners that we want to optimize and weight them by the area of the corresponding surface triangle.

$$E_\alpha = \sum_{t \in C_\alpha} E_t(a, b, c, d, e) \cdot A_{\phi^{-1}(t)} \tag{10}$$

In this optimization phase all layout edges are always tagged for alignment which ensures that all vertices of patch P_α are mapped into C_α. This energy only depends on five variables and is very well conditioned because of its geometric nature. Therefore we can use a simple and efficient Newton method to find a local minimum. Since the bilinear dependency is only an approximation we have to recompute the parameterization after each chart optimization. To initialize the charts we can simply use unit charts or the heuristic of [11]. The complete algorithm works as follows:

1. tag all layout edges for alignment
2. compute an initial parameterization with unit charts

3. iterate k times
   (a) optimize all charts individually
   (b) update transition functions
   (c) recompute parameterization
4. restore user-provided alignment tags and compute final parameterization

The bilinear relation is close to the exact dependency, therefore in all our experiments three iterations were sufficient to converge. Notice that our method is similar to [10]. However, instead of relaxing the layout vertices on the surface we relax them within the charts. This is more suitable for Reverse Engineering where the user provided layout is in general not allowed to be changed. In the next section we will discuss how to incorporate layout alignment constraints into the computation.

3.2 Alignment Constraints

The user can tag a subset of layout edges for alignment which ensures that it will be explicitly represented in the meshing. For the parameterization this means that the mapping of a tagged layout edge should be the straight line connecting both corresponding corners in the chart. Or in other words the parameter coordinates along the layout edge are not independent. The parameter coordinates at a point $p_e = (1-\lambda)p_i + \lambda p_j$ on the layout edge cannot be computed directly in the form $u_e = (1-\lambda)u_i + \lambda u_j$ because u_i and u_j are represented w.r.t different charts (see Figure 6 (a)). However by employing the transition function, we can express the alignment constraint in a simple form where the image of the layout edge is constrained to have the first coordinate equal to zero. This is exactly the lower right setting in Figure 4:

$$u_e^\gamma = (1-\lambda)SR_\alpha T_\alpha u_i^\alpha + \lambda R_\beta T_\beta u_j^\beta \stackrel{!}{=} \begin{pmatrix} 0 \\ * \end{pmatrix}. \tag{11}$$

The global linear system already has full rank, therefore after adding the alignment constraints we have to relax some other equations to be solved only in least squares

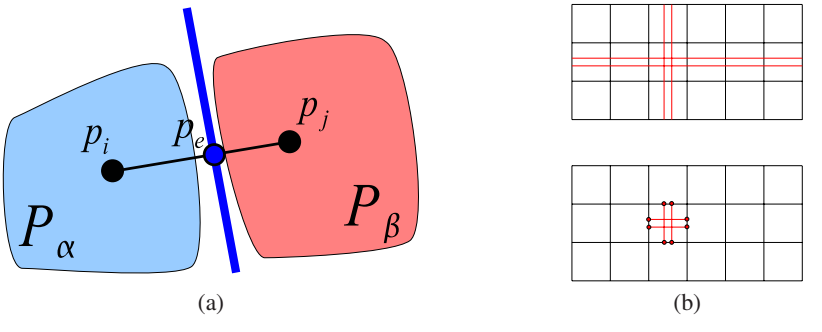


Fig. 6. (a) The parameter values at a layout edge can be computed at the intersection points of triangle mesh edges p_e by incorporating the transition function between both charts. (b) A local refinement of a quadrangular layout has global support (top). By allowing T-Vertices, the refinement of the layout remains local (bottom).

sense. A good choice are the harmonic constraints of all vertices which are involved in alignment constraints. This means pulling the parameterization onto the layout edge by allowing slight non-harmonicity near the constraint. Notice that our alignment constraints restrict only one coordinate of the parameterization and there is still a global relaxation in orthogonal direction. Finally the parameterization is formulated as a mixed least squares system of the form

$$\begin{bmatrix} A^T A & B^T \\ B & 0 \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} A^T b \\ c \end{pmatrix} \tag{12}$$

where the equations $Bx = c$ are fullfilled exactly and the equations $Ax = b$ are satisfied in a least squares sense. In the next section we will describe how to simplify the layout generation by allowing T-vertices.

3.3 T-Vertices

Restricting the layout graph to consist only of four sided polygons as done before is too restrictive in practice. Making a local refinement to keep more features, a global refinement would result as illustrated in Figure 6 (b). Remember that in many Reverse Engineering scenarios this layout is directly designed by a user and the effort should be as low as possible. Therefore we allow an arbitrary number of T-vertices per layout edge. This can be easily achieved by computing a separate transition function for each part of a layout edge. The parameter coordinates of T-vertices in a chart are defined by linear interpolation of the corners to keep the number of variables of a chart constant and allow to extract a mesh consisting only of quadrilaterals as explained in the next section.

4 Meshing

The meshing proceeds as follows, first a consistent quadmesh is constructed in the 2D charts of the parameterization which is then mapped to the surface. The four corners of a chart form a four sided polygon in the plane whereas each edge can be partitioned by T-vertices into several subintervals as depicted in Figure 7. By backmapping the chart polygon edge it is possible to compute the desired number of samples n_d which is the quotient of the length of the backmapped curve and the target edge-length for the meshing provided by the user. This value may be chosen differently for each layout edge. However in the case of a consistent quadmesh the number of samples cannot be chosen arbitrarily. There are the following consistency constraints:

1. The number of segments (quadmesh edges) on opposite edges of the chart polygon must be equal. In the example of Figure 7 (a) this means $n_1 + n_2 + n_3 = n_4 + n_5$ and $m_1 = m_2$.
2. Each T-vertex lies on a sampling position.
3. In each subinterval the samples are distributed linearly which guarantees that neighboring charts stitch together compatibly.

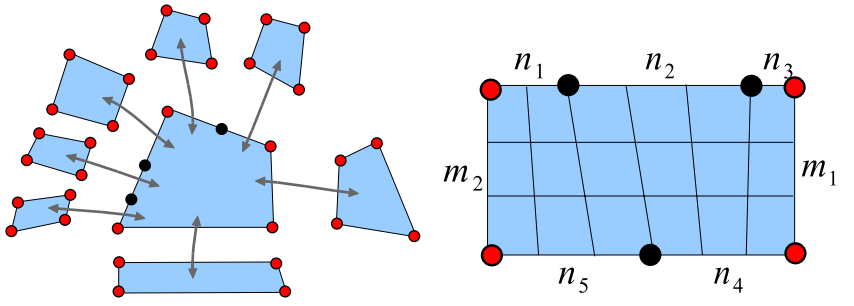


Fig. 7. (a) Allowing T-Vertices in the layout is possible in a simple way by computing a transition function per layout edge. (b) To get a closed quadrangulation the number of samples on opposite edges of a chart must be equal.

With these restrictions a consistent quadmesh can be constructed by connecting opposite sample pairs. This is always possible since condition 1 states that the number of samples is equal on opposing sides. We assemble the two equations per layout face in a common linear system $Bn = 0$ and compute free variables via Gauss elimination as done in [11]. Simply fixing the free variables by rounding the corresponding entries from the local desired number of samples n_d leads to poor results since the free variables computed by the Gauss elimination strongly depend on the numbering of the variables and can lead to strong deviations from the expected number of samples on other edges. Therefore we first compute the best continuous solution n_c which meets the constraint $Bn_c = 0$ and thus minimizes the deviation from the desired values n_d in a least squares sense. As a result we solve the linear system from equation 12 with $A = I_d$, $B = B$, $b = n_d$ and $c = 0$. Then rounding the free variables to the integer closest to the value of the continuous solution n_c leads to appealing results because the continuous solution captures the global necessary edge-length distribution.

5 Results and Discussion

In this section we will discuss the properties of the presented method by exploring some results. The first example is the sheared cube with unit edge length depicted in Figure 8 (a). This simple model illustrates the difference between the parameterization of [11] and our method which are displayed in (b) and (c) respectively. In (b) edge length distortions and S-shaped isolines are unavoidable because of the inherent tangential continuity of this method. This can be seen by unfolding neighboring faces of the cube where the isolines in the case of 8 (b) are smooth since the necessary curvature of the cone singularities is distributed over the whole geometry. In contrast to this result our method 8 (c) concentrates the tangential curvature at feature lines, i.e. regions of high geometric curvature where tangential continuity is not important. This example is indeed a hint how to use the presented method. Charts with five DOF's are advantageous for patches with a layout lying on geometric features while charts with two DOF's are

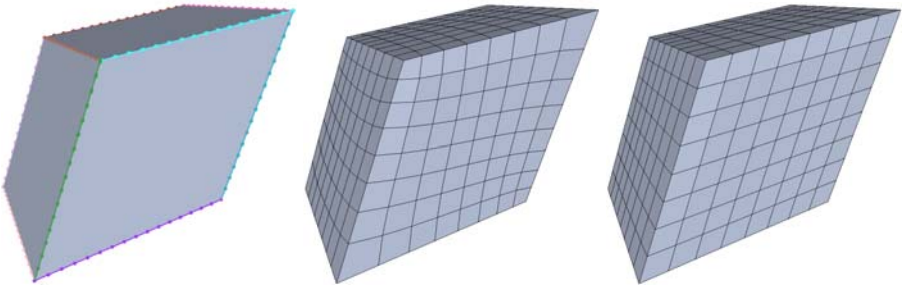


Fig. 8. (a) A sheared cube with unit edge length is segmented along the geometric edges. (b) Restricting to charts with 2 DOF's, length distortions and S-shaped isolines are unavoidable since the necessary curvature of the cone singularities is distributed over the whole geometry. (c) Our method concentrates the tangential curvature at geometric features where they don't influence the mesh quality. This approach leads to the expected result of uniformly shaped quadrilaterals.

better suited within smooth or flat regions. Typical objects consist of both types of regions, such that the user should be able to select for each patch which optimization is performed. This is possible in a straightforward way due to the fact that the optimization of individual charts is independent.

The second example is the car model depicted in Figure 2 and already discussed in the introduction. Figure 9 shows all chart polygons after 3 steps of optimization with 2 DOF's and 5 DOF's in (a) and (b) respectively. The presented optimization algorithm finds well shaped chart polygons robustly and produces almost symmetric configurations since the user-provided layout is almost symmetric. Compared to the time which is necessary for the solution of the global linear system, the optimization of charts is

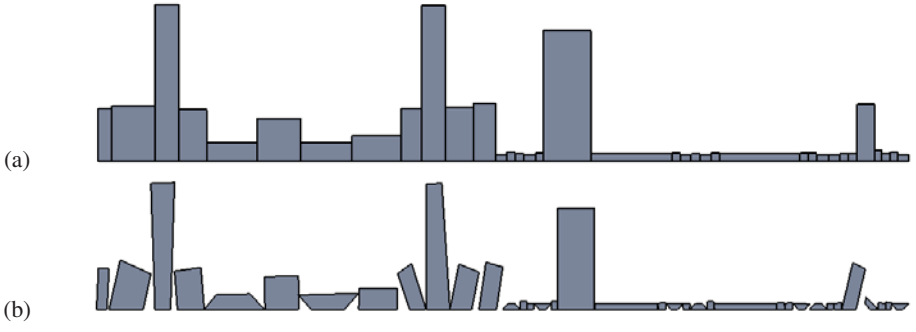
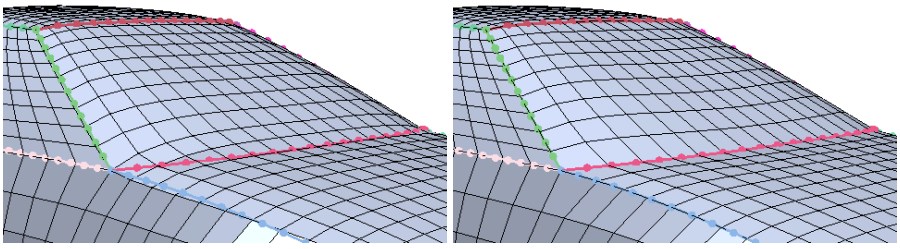


Fig. 9. (a) The corner positions of the car model's charts are optimized to lie on a rectangle. The resulting parameterization maximizes the isometry. (b) In this optimization the corners were allowed to lie in general position. Thus the resulting polygons are planar approximations of the surface patches.

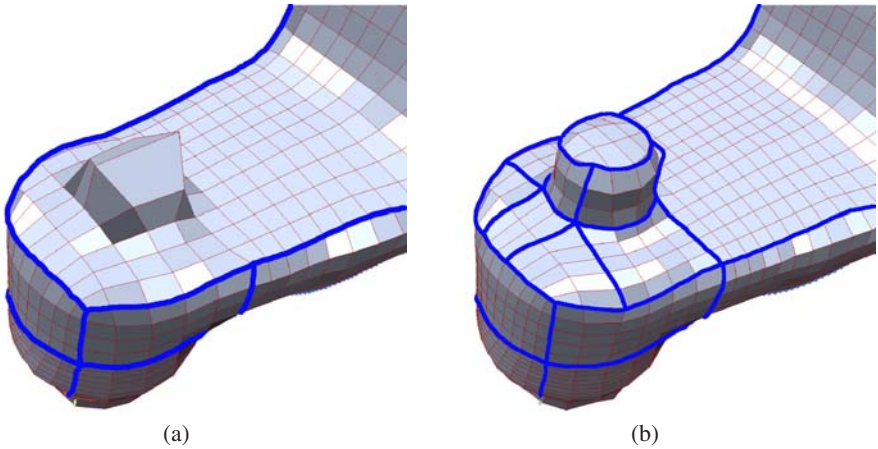


(a) (b)

Fig. 10. (a) The sharp corner between front window and hood is not represented in the globally smooth parameterization without alignment constrains. This leads to sampling artifacts, i.e. triangles that cut away the sharp corner. (b) The layout curve is tagged for alignment and consequently the mesh edges are pulled onto it, leading to a better approximation of the input geometry.

neglectable. Altogether the computation timings are comparable to [10] while in practice we need fewer iterations to converge. In all our examples we used the sparse direct solver SuperLU as proposed in [12].

In Figure 10 we demonstrate the usage of alignment constraints. Between the front window and the hood of the car is a sharp edge which should be represented in the final meshing to prevent sampling artifacts. Therefore the lower red layout curve is tagged for alignment. As one can see the isoline of the quadmesh connecting both endpoints of this layout curve in Figure 10 (a) is pulled onto the layout curve in Figure 10 (b) without introducing unnecessary distortion. However by using alignment constraints the computation time for solving the resulting mixed least-squares linear system is higher because of the doubled dimension.



(a) (b)

Fig. 11. (a) A rough layout (highlighted in blue) leads to large length distortions near a geometric feature. (b) T-vertices can be used to locally refine the layout with minimal effort. The new layout captures the geometric feature much better and avoids the length distortions.

The third example is the rockerarm model from Figure 1. For this mechanical part first a coarse layout was designed to guide the meshing. Figure 11 (a) shows a close up from the backside where large distortions appear near a geometric feature, not covered in the layout. In Figure 11 (b) the layout was locally refined by using T-vertices. In this way the designer can control the Reverse Engineering hierarchically by starting with a rough layout which is refined until all features are captured up to the desired quality. The overall effort to design a layout is minimized by using T-vertices.

6 Conclusion

In this paper, we have presented a global parameterization technique designed for the requirements of Reverse Engineering. Our solution is a generalization of well known global parameterization methods which allows to use chart corners in general position. To find good positions automatically an iterative algorithm is applied which optimizes the isometry of the resulting parameterization. This chart optimization can be used in combination with other methods as well. Additionally we described the incorporation of alignment constraints and T-vertices which are important ingredients of a Reverse Engineering solution.

Acknowledgments

This work has been supported by the UMIC Research Centre, RWTH Aachen University.

References

1. Alliez, P., Ucelli, G., Gotsman, C., Attene, M.: Recent advances in remeshing of surfaces. Research report, AIM@SHAPE Network of Excellence (2005)
2. Hormann, K., Lévy, B., Sheffer, A.: Mesh parameterization: theory and practice. In: SIGGRAPH 2007: ACM SIGGRAPH 2007 courses, p. 1. ACM, New York (2007)
3. Alliez, P., Cohen-Steiner, D., Devillers, O., Levy, B., Desbrun, M.: Anisotropic polygonal remeshing. ACM Transactions on Graphics. Special issue for SIGGRAPH conference, 485–493 (2003)
4. Marinov, M., Kobbelt, L.: Direct anisotropic quad-dominant remeshing. In: PG 2004: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference, Washington, DC, USA, pp. 207–216. IEEE Computer Society, Los Alamitos (2004)
5. Khodakovsky, A., Litke, N., Schröder, P.: Globally smooth parameterizations with low distortion. In: SIGGRAPH 2003: ACM SIGGRAPH 2003 Papers, pp. 350–357. ACM, New York (2003)
6. Ray, N., Li, W.C., Lévy, B., Sheffer, A., Alliez, P.: Periodic global parameterization. ACM Trans. Graph. 25(4), 1460–1485 (2006)
7. Kälberer, F., Nieser, M., Polthier, K.: Quadcover - surface parameterization using branched coverings. Computer Graphics Forum 26(3), 375–384 (2007)
8. Huang, J., Zhang, M., Ma, J., Liu, X., Kobbelt, L., Bao, H.: Spectral quadrangulation with orientation and alignment control. In: SIGGRAPH Asia 2008: ACM SIGGRAPH Asia 2008 Papers (2008) (to appear)

9. Ray, N., Vallet, B., Li, W.C., Lévy, B.: N-symmetry direction field design. ACM Transactions on Graphics (2008); Presented at SIGGRAPH
10. Dong, S., Bremer, P.T., Garland, M., Pascucci, V., Hart, J.C.: Spectral surface quadrangulation. In: SIGGRAPH 2006: ACM SIGGRAPH 2006 Papers, pp. 1057–1066. ACM, New York (2006)
11. Tong, Y., Alliez, P., Cohen-Steiner, D., Desbrun, M.: Designing quadrangulations with discrete harmonic forms. In: SGP 2006: Proceedings of the fourth Eurographics symposium on Geometry processing, Aire-la-Ville, Switzerland, pp. 201–210. Eurographics Association (2006)
12. Botsch, M., Bommes, D., Kobbelt, L.: Efficient linear system solvers for mesh processing. In: Martin, R., Bez, H.E., Sabin, M.A. (eds.) IMA 2005. LNCS, vol. 3604, pp. 62–83. Springer, Heidelberg (2005)

A Comparison of Three Commodity-Level Parallel Architectures: Multi-core CPU, Cell BE and GPU

André Rigland Brodtkorb[1] and Trond Runar Hagen[1,2]

[1] SINTEF, Dept. Appl. Math., P.O. Box 124, Blindern, N-0314 Oslo, Norway
[2] Centre of Mathematics for Applications (CMA),University of Oslo, Norway
{Andre.Brodtkorb,Trond.R.Hagen}@sintef.no

Abstract. We explore three commodity parallel architectures: multi-core CPUs, the Cell BE processor, and graphics processing units. We have implemented four algorithms on these three architectures: solving the heat equation, inpainting using the heat equation, computing the Mandelbrot set, and MJPEG movie compression. We use these four algorithms to exemplify the benefits and drawbacks of each parallel architecture.

1 Introduction

The gain in performance of computer programs has typically come from increased processor clock frequency and increased size of system memory. New computers have been able to handle larger problems in the same timeslot, or the same problem at greater speed. Recently, however, this trend has seemed to stop, and in the most recent years we have actually seen a *decrease* in clock frequency. The new trend is instead to increase the *number* of processor cores. There are several different multi-core designs in commodity hardware: your typical multi-core CPU consists of a few *fat* cores with a lot of complex logic; graphics processing units (GPUs) consist of several hundred *thin* processors with reduced functionality; and the Cell BE [1] consists of a mixture of fat and thin cores.

A key point to designing algorithms for these architectures is to understand their hardware characteristics. The aim of this work is to give a good understanding of how the hardware performs, with respect to a set of four example algorithms: solving the heat equation, inpainting missing pixels using the heat equation, computing the Mandelbrot set, and MJPEG movie compression. These four algorithms are chosen because they display different characteristics found in many real-world applications.

2 Related Work

There has been a lot of research into high-level programming abstractions to parallel architectures. All abstractions attempt to create an intuitive and simple API with as low as possible performance penalty. APIs such as OpenMP [2] and MPI [3] show good performance and have become de facto standards in shared memory and cluster systems, respectively. These two APIs have also been exploited to create abstractions to programming the Cell BE [4,5]. CellSs [6] is another abstraction for the Cell BE, consisting of a

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 70–80, 2010.

source to source compiler and runtime system. There have also been several high-level abstractions to programming the GPU for general purpose computation (GPGPU [7]). The most active languages today are CUDA [8] and Brook [9], where CUDA is a vendor specific language from NVIDIA, and Brook is an API with an extended version for AMD GPUs called Brook+ [10]. RapidMind [11] is a high-level C++ abstraction offering a common platform for different back-ends: multi-core CPUs, the Cell BE, and GPUs. This enables the programmer to easily run the same code on different hardware setups by simply changing the back-end. The back-ends themselves are responsible for low-level optimization and tuning, letting the programmer focus on high-level optimization of algorithms. OpenCL [12] is an API ratified by the Khronos group, in the same family of standards as OpenGL [13]. OpenCL offers a common low-level interface to program architectures such as multi-core CPUs, Cell BE and GPUs. Approved in December 2008, the first compilers have now appeared for GPUs, and the number of supported architectures is expected to rise. Such an open standard with support for multiple platforms is a great step in unifying heterogeneous programming efforts. In this work, we do not use or discuss the aforementioned higher-level abstractions, but instead present implementations created using lower-level tools for each architecture.

The algorithms we examine have been implemented earlier on all three architectures. The heat equation and other partial differential equations (PDEs) have been implemented on the Cell BE [14] and on the GPU [15]. Inpainting using PDE-based techniques has also been implemented on the GPU [16], but not, to our knowledge, on the Cell BE. A Mandelbrot generator is part of the NVIDIA CUDA SDK code samples, and writing an optimized Mandelbrot set generator for the Cell BE was the topic for the linux.conf.au 2008 hackfest. The main building block of MJPEG movie compression is also part of the NVIDIA CUDA SDK, and MJPEG 2000 has been implemented on the Cell BE [17]. We emphasize that our aim is not to compete with these implementations, but rather to show how implementations with a similar amount of optimization can uncover differences.

3 Architectures

Multi-core CPUs, the Cell BE, and GPUs all consist of several processors, with varying type and composition. This section briefly describes these architectures and the low-level languages and tools we have used to program them. We have chosen to compare models that were released in the same time period, and argue that the intrinsic differences we identify do not change dramatically with newer generations.

Modern CPUs incorporate a multi-core design that consists of multiple fat cores on a single chip, where each core has a large cache and a lot of logic for instruction level parallelism (ILP). More than half of the chip area is typically used for cache and logic, leaving a relatively small number of transistors for pure arithmetic operations. To utilize all cores in the multi-core processor we use OpenMP, a C, C++ and Fortran API for multi-threading in a shared memory system. In C++, the API consists of a set of compiler pragmas that for example can execute loops in parallel.

The Cell BE is a heterogeneous processing unit, consisting of one power processor element (PPE) and eight synergistic processing elements (SPEs). The PPE is a regular

fat CPU core, while the SPEs are thin cores capable of SIMD execution on 128 bits of properly aligned memory. The next version of the Cell BE is planned to have two PPEs and 32 SPEs. In the current version, each SPE has a small *local store* to hold their program and data, and is connected to the PPE through a fast on-chip interconnect. As opposed to the PPE, the SPEs have little or no ILP or cache, dedicating almost all transistors to floating point calculations. The API that we have used to access and program the SPEs is the SPE Runtime Management Library version 2 (libspe 2) [18]. In our use of libspe 2, the PPE program typically creates a context for each of the SPEs, and loads an SPE program (written in a subset of C++) into each context. The PPE then sets the SPEs to execute the programs. The SPEs are responsible for explicitly transferring data between local store and main memory using direct memory access (DMA). DMA executes asynchronously, meaning we can overlap memory transfer with computation. This effectively hides memory latency and transfer time. While the SPEs are computing, the PPE mainly acts as a supervisor handling support functions for the SPEs. This use is often referred to as the SPE-centric programming model, where the SPEs run the main part of the program.

The gaming industry is the main driving force behind the development of GPUs. Traditionally, the GPU only had a fixed set of graphics operations implemented in hardware, but recent generations have become programmable. Current high-end graphics cards have several hundred *stream processors* where almost all transistors are dedicated to floating point arithmetics. The NVIDIA GeForce 9800 GX2 for example, has 256 stream processors. The last generations of GPUs from NVIDIA can be programmed using CUDA, an extension to C that views the GPU as a general stream processor. CUDA typically uses the GPU to execute the same program, referred to as the *kernel*, over a large set (or stream) of data. To execute the kernel, the input data is first transferred to the GPU, and then the computational domain is divided into equally sized blocks. Each block has one virtual thread per element, and the blocks are computed independently. When the kernel has completed execution on all blocks, the data is typically read back to the CPU.

We have benchmarked the four algorithms on two different commodity level hardware setups. The CPU and GPU implementations were run on the same system, consisting of an Intel Core 2 Duo 2.4 GHz processor with 4 MiB cache, 4 GiB system memory, and an NVIDIA GeForce 8800 GTX card with 768 MiB graphics memory. The Cell BE implementations were run on a PlayStation 3, containing the 3.2GHz Cell BE processor with 6 available SPEs and 256 MiB system memory. The CPU has a theoretical performance of about 20 GFLOPS per core, and each SPE on the Cell BE has a theoretical performance of about 25 GFLOPS. The GPU has a theoretical performance of about 520 GFLOPS. All the implementations have been compiled using level three optimization and the *fast-math* option enabled. They also contain approximately the same amount of platform specific optimizations. The time and effort needed to program the different architectures, however, differs somewhat. Implementing the CPU version required less time and debugging efforts than the GPU version and the Cell BE version. The difference in effort comes from two main contributors: prior experience with the architecture and quality of programming tools, debuggers and profilers. If we attempt to disregard the impact of prior experience, we can give a few general remarks on

programming effort. Of the three architectures, we find that the CPU implementation uses the highest level API, and requires the least programming effort. Both the GPU and Cell BE implementations on the other hand, use lower level APIs that require detailed knowledge of the hardware to avoid performance pitfalls. Thus, they require more programming effort to reach the same level of optimization as the CPU.

4 Algorithms and Results

To compare the architectures, we have implemented four different algorithms: solving the heat equation with a finite difference method, inpainting missing pixels using the heat equation solver, computing the Mandelbrot set, and MJPEG movie compression. We have selected these algorithms because they exhibit different computational and memory access properties that are representative for a large range of real-world problems: the first two are memory streaming problems with regular and irregular memory access patterns, respectively, and the last two are number crunching problems with uniformly and nonuniformly distributed computation. All four algorithms can easily be parallelized and should thus fit the architectures well.

The following sections describe the algorithms and some details for the Cell BE and GPU implementations. Our CPU implementations use OpenMP to parallelize the execution, using a static scheduler for the two algorithms with a predetermined work-load and a dynamic scheduler for the two with a data-dependent work-load. We employ the `omp parallel for` pragma on the outmost loops, thus minimizing OpenMP overheads.

4.1 The Heat Equation

The heat equation describes how heat dissipates in a medium. In the case of a 2D homogeneous and isotropic medium, it can be written as $u_t = a(u_{xx} + u_{yy})$, where a is a material specific constant. Using an explicit finite difference scheme, the unknown solution $u_{i,j}^{n+1}$ in grid point (ih, jh) at time $(n+1)k$ is given by

$$u_{i,j}^{n+1} = a\frac{k}{h^2}\left(u_{i-1,j}^{n} + u_{i+1,j}^{n} + 4u_{i,j}^{n} + u_{i,j-1}^{n} + u_{i,j+1}^{n}\right), \tag{1}$$

where h and k determine the spatial and temporal resolution, respectively.

Characteristics - The heat equation benchmark shows how efficient each architecture is at streaming data. Solving the heat equation using this explicit finite difference scheme requires approximately the same number of memory accesses as arithmetic instructions. As the processor clock speeds are much higher than the memory clock speeds on our architectures, the performance should be limited by memory bandwidth. The memory access pattern, however, is regular and thus enables efficient hardware pre-fetching techniques on the multi-core CPU. Thus, the multi-core CPU should get full use of its large cache as long as the two cores keep from fighting over the same set of cache lines. On the Cell BE we can try to hide memory access by overlapping this with computation, and on the GPU explicitly gather the data into fast on-chip memory called *shared memory*.

Listing 1.1. CellHeat.cpp

```
for (i=1; i<=height; ++i) {
  //Increment front, back, up, center, down, next
  DMA_REQUEST_ROW_FROM_MEMORY(input[next]);
  DMA_WAIT(input[down]);
  DMA_WAIT(output[front]);
  for(j=1; j<width-1; ++j)
    output[front][j] = 0.125f * (input[up][j]
                                 + input[center][j-1]
                                 + 4.0f*input[center][j]
                                 + input[center][j+1]
                                 + input[down][j]);
  DMA_REQUEST_ROW_TO_MEMORY(output[front]);
}
```

This algorithm displays properties found when solving other PDEs using explicit finite difference/element/volume methods. Image filtering, such as convolution, is another class of algorithms with the same properties.

Implementation - By examining the computational molecule for $u_{i,j}^{n+1}$, we see that it depends on the value at grid-point (ih, jh) and the four closest neighbours in the previous time-step. Because of this dependency, we cannot simply update the value of $u_{i,j}^{n}$ in-place. By having two buffers, we store the result of odd time-steps in the "odd" buffer, reading from the "even" buffer, and conversely for even time-steps. This way we can update all elements in parallel at each time-step.

On the Cell BE, we divide the domain into blocks with an equal number of rows, and let each SPE calculate its sub-domain. Listing 1.1 shows the Cell BE algorithm in pseudo code, and Figure 1 is a graphical representation. For each row we want to compute in the output, we need to explicitly gather the needed data from main memory to local store and write the result back. Using asynchronous memory transfer, we can overlap with computation, and only keep four rows of input in local store at any given time. We have used intrinsic SIMD instructions in the actual implementation, as the SPEs are SIMD processors.

On the GPU, we divide the computational domain into blocks with one virtual thread per element. For each block, each thread reads one element from the GPUs main memory into shared memory. The shared memory is fast on-chip memory accessible to all

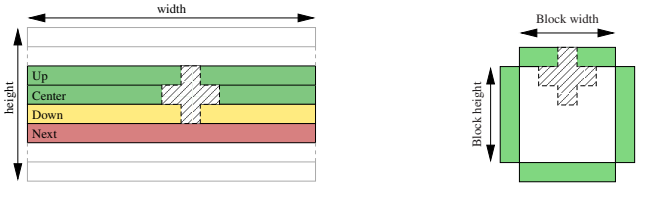


Fig. 1. The heat equation computed on the Cell BE (left) and on the GPU (right)

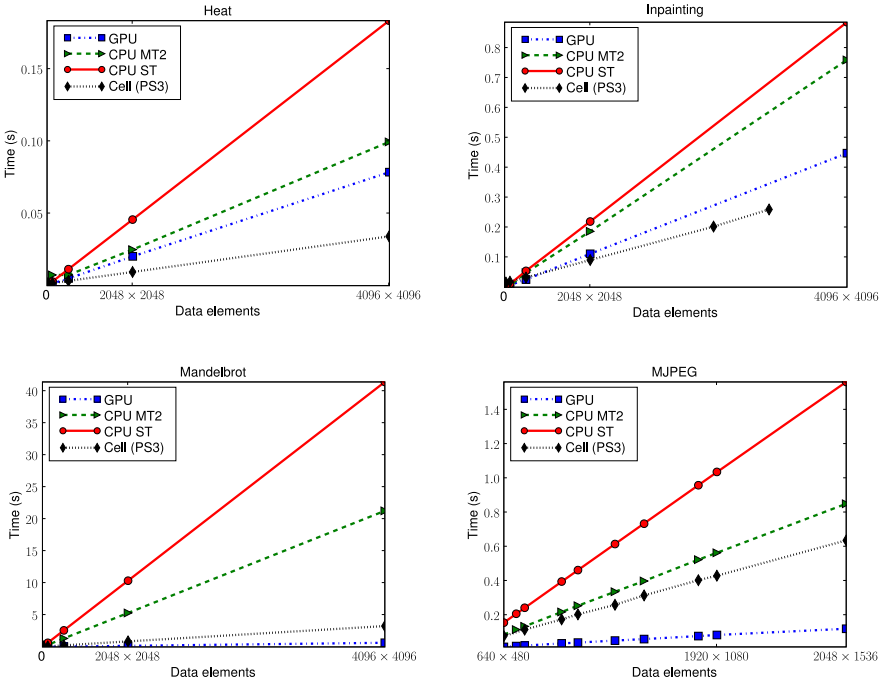


Fig. 2. Runtime of the four different algorithms: the heat equation (top left), inpainting (top right), the Mandelbrot set (bottom left), MJPEG (bottom right). CPU ST represents the serial CPU version, and CPU MT2 represents the multi-threaded CPU version running on two CPU cores.

virtual threads in the same block. We also have to read the *apron* (also called ghost cells) into shared memory. The apron consists of the data-elements outside the block that we need for our computational molecule. Figure 1 shows the block and its apron. When all data has been read into shared memory, each thread computes one output element and writes the result to GPU main memory.

Results - Figure 2 shows the runtime of computing one step of the heat equation on the different architectures. For the GPU, this includes the time it takes to read the result back from graphics memory. The other two architectures place the result in main memory without the explicit read-back. Table 1 breaks the GPU and Cell BE run-times up into time spent computing and time spent waiting for memory transfers to complete. The table shows that the GPU suffers from an expensive read-back of data from the GPU to the CPU, whilst the Cell BE is able to hide much of the memory reads and writes by overlapping with computation. This explains why the GPU barely beats the multi-core CPU implementation. The GPU, however, will outperform the other architectures if the data is allowed to remain on the GPU throughout multiple time-steps.

Table 1. Breakdown of running times. For the GPU, the percentages represent kernel execution time and GPU-CPU transfer time. For the Cell BE, the time represents time spent computing, and stall time while waiting for DMA requests to complete.

		Heat	Inpainting	Mandelbrot	MJPEG
GPU	Memory	75%	55%	0%	80%
	Computation	25%	45%	100%	20%
Cell BE	Memory	10%	10%	0%	5%
	Computation	90%	90%	100%	95%

4.2 Inpainting

Noisy images (e.g., from poor television reception) can be repaired by *inpainting*. Here we use the heat equation on masked areas as a naïve example of inpainting. Using this approach, information from the area surrounding a block of noisy pixels will be diffused into the block and fill in the missing values. Technically, each masked element is updated using Equation (1), whereas we set $u_{i,j}^{n+1} = u_{i,j}^{n}$ for unmasked elements.

Characteristics - The inpainting benchmark shows how efficient each architecture is at streaming data, executing conditionals, and computing on a subset of the data. Inpainting using the heat equation requires even fewer arithmetic instructions per memory access than solving the heat equation. This is because we have to read the mask for all elements, but only run computations on a few of them. This memory access pattern will further make matters worse, as the CPU will underutilize its cache. The GPU will have problems with the added conditional as all processors in one *warp* (currently 32 stream processors) are forced to run commands synchronously. The effect is that even if only one processor in the warp branches differently from the others, the whole warp must evaluate both sides of the branch.

This algorithm has properties that are also found in other algorithms with a lot of memory access and little computation, e.g., image processing algorithms working on masked areas.

Implementation - Both the Cell BE and GPU implementations of this algorithm are very similar to the heat equation solver. The major difference is that the Cell BE and GPU must explicitly gather the mask into local store and shared memory, respectively.

Results - For each image resolution, we used a noise mask to inpaint approximately 5% of the image using ten time-steps of the heat equation. Figure 2 shows that the GPU performs relatively better compared to the heat equation solver. This is because the GPU is able to keep the data in graphics memory throughout the ten passes. This lessens the effect of the single read-back, as shown in Table 1. Because the GPU executes one warp synchronously, it uses about the same time to complete one pass of inpainting, as it does to complete one pass of the whole heat equation. The Cell BE computes four elements synchronously using intrinsic SIMD instructions. However, it only runs slightly faster per pass than the heat equation. This can be explained by the fact that the SPEs do not contain a conditional branch predictor, making the branch almost as expensive as

computing the heat equation itself. Using a computationally more demanding algorithm would diminish the effect of the branch on the Cell BE. There was also not enough physical system memory to benchmark domains larger than approximately 3600×3600 for the Cell BE. The single threaded CPU version runs much faster per pass than the heat equation, whilst the multi-threaded version only has a marginal speedup. This can be explained by the increased load on the memory bus, as the additional mask has to be loaded into cache, compared to the heat equation solver alone. Using multiple cores does not increase performance when the bottleneck is the memory bandwidth.

4.3 The Mandelbrot Set

The Mandelbrot set is a fractal that has a very simple recursive definition:

$$M = \left\{ c \in \mathbb{C} : z_0 = c, \quad z_{n+1} = z_n^2 + c, \quad \sup_{n \in \mathbb{N}} |z_n| < \infty \right\}. \tag{2}$$

Informally, it is the set of all complex numbers that do not tend towards infinity when computing z_{n+1}. When computing the Mandelbrot set, one uses the fact that c belongs to M if and only if $|z_n| < 2$ for all $n \geq 0$. Typically, one picks a set of discrete complex points C and a fixed m, and then the point $c \in C$ is assumed to be in the set if $|z_n| < 2$ for all $n \leq m$.

Characteristics - This benchmark shows how well each architecture performs floating point operations, and how it copes with dynamic workloads. Computing whether a complex coordinate belongs to the Mandelbrot set requires a lot of computation, and only a single memory write. For coordinates that are in the set the program has to compute all m iterations while coordinates outside often compute only a few. This means that neighbouring pixels often have very different workloads, as the boundary has a highly complex geometry.

Computing the Mandelbrot set exhibits properties also found in algorithms such as ray-tracing and ray-casting, as well as many other iterative algorithms. Algorithms with a lot of computation per memory access, such as protein folding, also show these properties.

Implementation - Using the abscissa as the real part and the ordinate as the imaginary part we create a set of complex numbers C. For each point $c \in C$, a while-loop computes z_n until $|z_n| > 2$ or $n > m$. The pixel is colored using the value of n when the loop terminates, yielding a gray-scale image where all white pixels are assumed to be part of the Mandelbrot set.

In our Cell BE implementation, we partition the domain into lines, where only the real part of c varies between pixels. We know that the computational time for each line can differ drastically, so we use a dynamic load distribution. The PPE simply fills a fixed length queue with line-numbers for each SPE, and the SPEs then start processing their queue. The PPE continues to enqueue line-numbers in non-full queues until the whole domain has been enqueued. The actual computation is done using SIMD instructions to utilize the hardware.

Results - Table 1 shows that computing whether a complex coordinate belongs to the Mandelbrot set or not is computationally intensive, and Figure 2 shows that both the GPU and Cell BE perform very well. Using multiple CPU cores scales perfectly. On the GPU, it does not drastically affect the results that two pixels close to each other can have very different workloads. Even though each warp is executed synchronously, most warps simply contain only pixels within the set, or outside it. Thus, the number of warps with a mixture of pixels within and outside the set is often far less than the number of warps with a relatively homogeneous workload.

4.4 MJPEG

MJPEG is an "industry standard" for compression of a movie stream. The main part of the algorithm consists of dividing each frame into 8×8 blocks, computing the discrete cosine transform (DCT), and then quantizing each block:

$$p_{u,v} = \alpha(u)\alpha(v) \sum_{i=0}^{7} \sum_{j=0}^{7} g_{i,j} \cos\left[\frac{\pi}{8}(i+0.5)u\right] \cos\left[\frac{\pi}{8}(j+0.5)v\right],$$

$$\alpha(n) = \begin{cases} \sqrt{1/8}, n = 0 \\ \sqrt{2/8}, n \neq 0 \end{cases},$$

$$r_{u,v} = \text{round}\left(p_{u,v}/q_{u,v}\right).$$

Here, $g_{i,j}$ is the element (i, j) from the 8×8 block, $p_{u,v}$ is the amplitude of frequency (u, v), $q_{u,v}$ is element (u, v) of the quantization matrix, and $r_{u,v}$ is the result.

Characteristics - The MJPEG results show how efficient each architecture is with a typical data flow, where the computationally intensive part of the code is accelerated, and the rest of the code runs on a single CPU core. Computing the DCT and then quantizing is an algorithm that is both cache friendly and requires a lot of computation per memory access. The performance should therefore be limited by processing power rather than memory access. Computing the cosines is typically also very expensive.

The MJPEG algorithm is representative for many image and movie compression algorithms, as they all do a lot of computation per memory access. These properties are also displayed in computationally heavy image processing algorithms, such as computing the FFT and image registration.

Implementation - To optimize for DMA transfer, our Cell BE version computes a row of blocks. Since the total amount of work is constant in each block-line, we use a static load distribution, dividing the domain into an equal number of block-lines per SPE. The SPE then computes each block in each block-line until there are no more block-lines to compute.

MJPEG compression fits the GPU perfectly, as CUDA already assumes that computation should be split up into blocks. We simply use a block-size of 8×8 and let CUDA automatically schedule the blocks in our GPU implementation.

Results - Figure 2 shows the time it takes to compute the DCT, quantize, and Huffman code one image consisting of an intensity channel and two chrominance channels with half the horizontal and vertical resolution of the intensity image. Table 1 breaks down the time spent on DCT and quantization, as these are the parts that have been accelerated. The GPU is heavily penalized for overhead connected with starting each memory transfer. It has to upload the three images and download them again after the computations. The overhead with starting these six memory transfers is substantial. The Cell BE does not suffer from such overheads, but is able to overlap almost all memory access by computation. However, Huffman coding using the PPE is very slow. The PPE takes 50% more time to complete compared to the CPU, even though it is the exact same code.

5 Summary

We have examined how a set of four algorithms perform on three sets of commodity level parallel hardware. All the algorithms we displayed, except inpainting, scaled well on the CPU. The inpainting algorithm saturates the memory bus, which again limits performance. The Cell BE performs well on algorithms where memory access can be hidden by computations, and when it comes to raw floating point performance. However, it has a relatively slow PPE core that can limit performance. The limited system memory on the PlayStation 3 can certainly also be a problem. The GPU is, by far, the best performing architecture for computationally intensive operations, but transferring memory to and from the graphics card can be very expensive.

This work has focused on four applications that show different properties of the three architectures. This is a small comparison, and it would be of great use to broaden the number of algorithms, architectures, and programming languages to give a broader understanding of commodity level parallel architectures.

The authors would like to thank the anonymous reviewers for their thorough comments and feedback.

References

1. IBM, Sony, Toshiba: Cell Broadband Engine programming handbook version 1.1 (2007)
2. OpenMP Architecture Review Board: OpenMP application program interface version 2.5 (2005)
3. Message Passing Interface Forum: MPI-2: Extensions to the message-passing interface (2003)
4. Ohara, M., Inoue, H., Sohda, Y., Komatsu, H., Nakatani, T.: MPI microtask for programming the Cell Broadband Engine processor. IBM Systems Journal 45, 85–102 (2006)
5. Eichenberger, A., O'Brien, J., O'Brien, K., Wu, P., Chen, T., Oden, P., Prener, D., Shepherd, J., So, B., Sura, Z., Wang, A., Zhang, T., Zhao, P., Gschwind, M.: Optimizing compiler for the Cell processor. In: Intl. Conf. on Parallel Architectures and Compilation Techniques, Washington, DC, USA, pp. 161–172. IEEE Computer Society, Los Alamitos (2005)
6. Bellens, P., Perez, J., Badia, R., Labarta, J.: CellSs: a programming model for the Cell BE architecture. In: SuperComputing 2006 (2006)

7. Owens, J., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A., Purcell, T.: A survey of general-purpose computation on graphics hardware. Computer Graphics Forum 26, 80–113 (2007)
8. NVIDIA corporation: NVIDIA CUDA programming guide version 1.1 (2007)
9. Buck, I., Foley, T., Horn, D., Sugerman, J., Houston, M., Hanrahan, P.: Brook for GPUs: Stream computing on graphics hardware (2004)
10. AMD Corporation: AMD stream computing revision 1.3.0 (2008)
11. McCool, M.: Data-parallel programming on the Cell BE and the GPU using the rapidmind development platform. In: GSPx Multicore Applications Conference (2006)
12. Khronos OpenCL Working Group: The OpenCL specification 1.0 (2008)
13. Shreiner, D., Woo, M., Neider, J., Davis, T.: OpenGL Architecture Review Board. OpenGL Programming Guide: The Official Guide to Learning OpenGL, 6th edn. Addison-Wesley, Reading (2007)
14. Williams, S., Shalf, J., Oliker, L., Kamil, S., Husbands, P., Yelick, K.: The potential of the Cell processor for scientific computing. In: Computing Frontiers 2006 (2006)
15. Hagen, T., Henriksen, M., Hjelmervik, J., Lie, K.A.: How to solve systems of conservation laws numerically using the graphics processor as a high-performance computational engine. In: Geometric Modelling, Numerical Simulation and Optimization: Industrial Mathematics at SINTEF. Springer, Heidelberg (2007)
16. Hagen, T., Rahman, T.: GPU-based image inpainting using a TV-Stokes equation (2008) (preprint)
17. Muta, H., Doi, M., Nakano, H., Mori, Y.: Multilevel parallelization on the Cell / B.E. for a motion JPEG 2000 encoding server. In: MULTIMEDIA 2007 (2007)
18. IBM: Software development kit for multicore acceleration version 3.0 (2007)

Mean Distance from a Curve to Its Control Polygon

Jesús Carnicer[1,⋆] and Jorge Delgado[2,⋆]

[1] Dept. Applied Mathematics/IUMA, University of Zaragoza, 50009 - Zaragoza, Spain
carnicer@unizar.es
[2] Dept. Applied Mathematics, Escuela Politécnica Universitaria de Teruel, University of Zaragoza, 44003 - Teruel, Spain

Abstract. The mean distance between a curve and its control polygon is bounded in terms of the norm of second order differences of the control points. We also analyze the distance of a rational curve to its control polygon and suggest a choice of the weight for obtaining rational curves much closer to its control polygon than Bézier curves.

Keywords: Control polygon, rational curve, mean distance.

1 Introduction

In Computer-Aided Geometric Design, parametric curves

$$\gamma(t)=\sum_{i=0}^{n} P_i u_i(t), \quad t\in[a,b], \tag{1}$$

are generated by *blending systems* of functions $(u_0,\dots,u_n)$, $u_i:[a,b]\to\mathbb{R}$, $u_i(t)\geq 0$, $t\in[a,b]$, $i=0,\dots,n$, $\sum_{i=0}^{n} u_i(t)=1$, $t\in[a,b]$. The points $P_i\in\mathbb{R}^k$, $i=0,\dots,n$, are called *control points* of γ and $P_0\cdots P_n$ is the *control polygon* of the curve.

When a parametric curve (1) is generated by a blending system of n functions $(u_0,\dots,u_n)$, the *convex hull property* is satisfied, that is, the curve lies in the convex hull of its control polygon. Another important property is *linear precision*

$$\sum_{i=0}^{n} \alpha_i u_i(t)=t, \quad t\in[a,b], \tag{2}$$

for a strictly increasing sequence of real numbers $(\alpha_i)_{i=0}^n$. If a system of functions $(u_0,\dots,u_n)$ has linear precision, the graph $(t,f(t))$, $t\in[a,b]$, of the real function $f(t)=\sum_{i=0}^{n} c_i u_i(t)$ is the parametric curve (1) with control points $P_i=(\alpha_i,c_i)^T$ for all $i\in\{0,1,\dots,n\}$. Both the Bernstein basis and the B-spline basis have the linear precision property.

The *endpoint interpolation property* is also usually required: the first control point always coincides with the first point of the curve and the last control point always

⋆ Research partially supported by the Spanish Research Grant MTM2009-07315, by Gobierno de Aragón and Fondo Social Europeo.

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 81–92, 2010.

coincides with the last point of the curve. If a system $(u_0,\dots,u_n)$ satisfying the endpoint interpolation property has linear precision, then $\alpha_0 = a$ and $\alpha_n = b$.

We are interested in providing a measure of the degree of approximation of a parametric curve by its control polygon. This problem has been previously discussed in several papers (c.f. [1,5,6,7]). For this purpose, we will bound the mean distance between a parametric curve and its control polygon. A natural way to parameterize the polygon is by the piecewise linear function $\pi : [a,b] \to \mathbb{R}^k$, defined by

$$\pi(t) = \frac{\alpha_i - t}{\alpha_i - \alpha_{i-1}} P_{i-1} + \frac{t - \alpha_{i-1}}{\alpha_i - \alpha_{i-1}} P_i, \quad \alpha_{i-1} \le t \le \alpha_i.$$

We can also represent π as a parametric curve with control points P_i, $i = 0,\dots,n$ using the first degree B-spline basis

$$\pi(t) = \sum_{i=0}^{n} P_i N_i(t), \quad N_i(t) := \begin{cases} (t-\alpha_{i-1})/(\alpha_i - \alpha_{i-1}), & \text{if } \alpha_{i-1} \le t \le \alpha_i, \\ (\alpha_{i+1}-t)/(\alpha_{i+1} - \alpha_i), & \text{if } \alpha_i \le t \le \alpha_{i+1}, \\ 0, & \text{elsewhere,} \end{cases} \tag{3}$$

for all $i \in \{0,\dots,n\}$, where $\alpha_{-1} := a$ and $\alpha_{n+1} := b$. Therefore the curve γ and its control polygon can both be regarded as two parametric curves generated by the same control points, but different blending system of functions.

Along this paper we will adhere to the notation in [1]: $\nabla c_i := c_i - c_{i-1}$ denotes the usual backward difference and $\delta c_i := \nabla c_i / \nabla \alpha_i - \nabla c_{i-1} / \nabla \alpha_{i-1}$ is a normalized second order backward difference. Observe that $\sum_{i=0}^{n} c_i N_i(t)$ is a convex function if and only if $\delta c_i \ge 0$, $i = 2,\dots,n$.

A system of functions $(u_0,\dots,u_n)$ satisfying the linear precision property (2) is *convexity preserving* if for any set of coefficients $c_0,\dots,c_n$ such that $\delta c_i \ge 0$ for all $i = 2,\dots,n$, the function $\sum_{i=0}^{n} c_i u_i(t)$ is convex.

In Section 2, the mean distance between two curves with the same control polygon is bounded and the result is applied to bound the mean distance between a curve and its control polygon. In Section 3, we discuss how to apply the above results to rational curves. A suggestion for the choice of the weights is provided.

2 Mean Distance between Two Parametric Curves

For a given norm $\|\cdot\|$ in $\mathbb{R}^k$, we will bound

$$M = \frac{1}{b-a} \int_a^b \|\gamma(t) - \bar{\gamma}(t)\| dt,$$

which can be regarded as the mean distance between two parametric curves γ, $\bar{\gamma}$. Let us observe that the total area between both curves has a more complicated formula

$$\frac{1}{2} \int_a^b \int_0^1 \|(\gamma(t) - \bar{\gamma}(t)) \times ((1-s)\gamma'(t) + s\bar{\gamma}'(t))\|_2 \, ds \, dt.$$

However, if L and $\bar{L}$ are the lengths of γ and $\bar{\gamma}$, $M(L+\bar{L})/2$ can be regarded as a rough estimation of the total area between both curves.

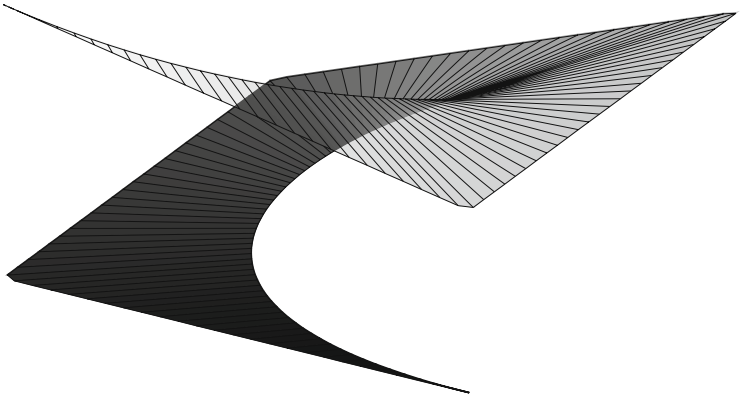


Fig. 1. Distance and area between a Bézier curve and its control polygon

Let $(u_0,\dots,u_n)$ and $(\bar{u}_0,\dots,\bar{u}_n)$ be two blending systems defined on an interval $[a,b]$ having the linear precision property for some common strictly increasing sequence $(\alpha_i)_{i=0}^n$. We shall assume that both curves

$$\gamma(t)=\sum_{i=0}^{n} P_i\,u_i(t) \quad \text{and} \quad \bar{\gamma}(t)=\sum_{i=0}^{n} P_i\,\bar{u}_i(t) \tag{4}$$

share the same control polygon.

In Theorem 1, we bound $\int_a^b \|\gamma(t)-\bar{\gamma}(t)\|w(t)dt$, where $w(t)$ is a weight function, usually taken as 1. Similary as in [1], we will express the bound as a constant times $\max_{\{i=2,\dots,n\}} \|\delta P_i\|/d_i$. This constant is of the form $\int_a^b D(t)w(t)dt$, where $D(t)$ is a function whose computation can be considerably simplified in the particular case where $\bar{\gamma}(t)$ is the control polygon $\pi(t)$. The result has been stated with full generality in order to be applied to Bézier, B-splines, rational and other kinds of curve representations without increasing the complexity of the proof.

Theorem 1. *Let $(u_0,\dots,u_n)$ and $(\bar{u}_0,\dots,\bar{u}_n)$ be two blending systems defined on an interval $[a,b]$. Let us assume that*

$$\sum_{i=0}^{n} \alpha_i\,u_i(t)=t \quad \textit{and} \quad \sum_{i=0}^{n} \alpha_i\,\bar{u}_i(t)=t$$

for some common strictly increasing sequence of values $\alpha_0,\dots,\alpha_n$ and let $w:[a,b]\to[0,+\infty)$ be a nonnegative weight function. Let $\gamma(t)$ and $\bar{\gamma}(t)$ be two parametric curves generated by the same polygon $P_0\cdots P_n$, with $P_0,\dots,P_n\in\mathbb{R}^k$. Then, for any sequence $d_2,\dots,d_n$ of positive numbers we have

$$\int_a^b \|\gamma(t)-\bar{\gamma}(t)\|\,w(t)dt \le \max_{i\in\{2,\dots,n\}} \frac{\|\delta P_i\|}{d_i} \int_a^b D(t)w(t)dt \tag{5}$$

where

$$D(t):=\sum_{i=2}^{n} d_i\,|v_i(t)-\bar{v}_i(t)|$$

and

$$v_i(t) := \sum_{j=i}^{n} (\alpha_j - \alpha_{i-1}) u_j(t), \quad \bar{v}_i(t) := \sum_{j=i}^{n} (\alpha_j - \alpha_{i-1}) \bar{u}_j(t).$$

If, in addition, $v_i(t) \geq \bar{v}_i(t)$ for all $t \in [a,b]$, then

$$D(t) = \bar{s}(t) - s(t), \text{ where } s(t) := \sum_{i=0}^{n} \sigma_i u_i(t),\ \bar{s}(t) := \sum_{i=0}^{n} \sigma_i \bar{u}_i(t),$$

and $(\sigma_i)_{i=0}^n$ is the sequence uniquely defined by

$$\sigma_0 = \sigma_n = 0, \quad \delta\sigma_i = -d_i, \text{ for all } i \in \{2, \ldots, n\}.$$

Proof. Taking into account that

$$\gamma(t) - \bar{\gamma}(t) = \sum_{i=0}^{n} P_i (u_i(t) - \bar{u}_i(t)) = \sum_{i=1}^{n} \nabla P_i \sum_{k=i}^{n} (u_k(t) - \bar{u}_k(t)) =$$
$$\sum_{i=1}^{n} \frac{\nabla P_i}{\nabla \alpha_i} \nabla \alpha_i \sum_{k=i}^{n} (u_k(t) - \bar{u}_k(t)) = \sum_{i=2}^{n} \delta P_i \sum_{j=i}^{n} \sum_{k=j}^{n} \nabla \alpha_j (u_k(t) - \bar{u}_k(t))$$

we obtain

$$\gamma(t) - \bar{\gamma}(t) = \sum_{i=2}^{n} \delta P_i (v_i(t) - \bar{v}_i(t))$$

and

$$\int_a^b \|\gamma(t) - \bar{\gamma}(t)\| w(t) dt = \int_a^b \Big\| \sum_{i=2}^{n} \delta P_i (v_i(t) - \bar{v}_i(t)) \Big\| w(t) dt.$$

Then we deduce from the last formula that

$$\begin{aligned} \int_a^b \|\gamma(t) - \bar{\gamma}(t)\| w(t) dt &\leq \sum_{i=2}^{n} \|\delta P_i\| \int_a^b |v_i(t) - \bar{v}_i(t)| w(t) dt \\ &= \sum_{i=2}^{n} \frac{\|\delta P_i\|}{d_i} \int_a^b d_i |v_i(t) - \bar{v}_i(t)| w(t) dt \end{aligned}$$

and (5) follows. If, in addition, $v_i(t) \geq \bar{v}_i(t)$ for all $t \in [a,b]$, we can write

$$\sum_{i=2}^{n} d_i |v_i(t) - \bar{v}_i(t)| = \Big(\beta(t - \alpha_0) - \sum_{i=2}^{n} d_i \bar{v}_i(t)\Big) - \Big(\beta(t - \alpha_0) - \sum_{i=2}^{n} d_i v_i(t)\Big)$$

where $\beta = (\alpha_n - \alpha_0)^{-1} \sum_{l=2}^{n} (\alpha_n - \alpha_{l-1})$. Now let us prove that

$$s(t) = \beta(t - \alpha_0) - \sum_{i=2}^{n} d_i v_i(t).$$

By the linear precision property and by the definition of v_i, we have that

$$\begin{aligned} \beta(t - \alpha_0) - \sum_{i=2}^{n} d_i v_i(t) &= \beta \sum_{i=1}^{n} (\alpha_i - \alpha_0) u_i(t) \\ &\quad - \sum_{i=2}^{n} \sum_{l=2}^{i} d_l (\alpha_i - \alpha_{l-1}) u_i(t). \end{aligned}$$

We can easily check that the coefficient of u_0 in the previous expression is 0 and that the coefficient of u_n is $\beta(\alpha_n-\alpha_0)-\sum_{l=2}^{n} d_l(\alpha_n-\alpha_{l-1})=0$. We can also verify that $\delta\left[\beta(\alpha_i-\alpha_0)-\sum_{l=2}^{i} d_l(\alpha_i-\alpha_{l-1})\right]=-d_i$. Hence $\beta(\alpha_i-\alpha_0)-\sum_{l=2}^{i} d_l(\alpha_i-\alpha_{l-1})=\sigma_i$, $i=0,\ldots,n$. Analogously we can prove that

$$\bar{s}(t)=\beta(t-\alpha_0)-\sum_{i=2}^{n} d_i\,\bar{v}_i(t),$$

and hence $D(t)=\bar{s}(t)-s(t)$. □

We observe that, if $d_i=\alpha_i-\alpha_{i-2}$, $i=2,\ldots,n$ then $\sigma_i=(\alpha_i-a)(b-\alpha_i)$, $i=0,\ldots,n$ and we can also express $\bar{s}(t)-s(t)$ in Theorem 1 in terms of the operator

$$E[f](t):=\sum_{i=0}^{n} f(\alpha_i)(u_i(t)-\bar{u}_i(t)).$$

by means of the formula $\bar{s}(t)-s(t)=E[(t-a)(b-t)]$.

Now let us consider the particular case where $(\bar{u}_0,\ldots,\bar{u}_n)=(N_0,\ldots,N_n)$, and hence $\bar{\gamma}(t)=\pi(t)$ is the control polygon of $\gamma(t)$.

Corollary 2. *Let $(u_0(t),\ldots,u_n(t))$ be a blending basis defined on $[a,b]$ such that there exists $\alpha_0<\alpha_1<\cdots<\alpha_n$ with $\sum_{i=0}^{n}\alpha_i u_i(t)=t$ and let $d_2,\ldots,d_n$ be a sequence of positive numbers. Then, for any parametric curve $\gamma(t)=\sum_{i=0}^{n} P_i u_i(t)$ and the parameterization $\pi(t)$ of its corresponding control polygon given by (3), we have*

$$\int_a^b \|\gamma(t)-\pi(t)\|\,dt\le \max_{i\in\{2,\ldots,n\}}\frac{\|\delta P_i\|}{d_i}\int_a^b D(t)\,dt$$

where

$$D(t):=\sum_{i=2}^{n} d_i(v_i(t)-\bar{v}_i(t))=\bar{s}(t)-s(t), \tag{6}$$

$$v_i(t):=\sum_{j=i}^{n}(\alpha_j-\alpha_{i-1})u_j(t),\quad \bar{v}_i(t):=\sum_{j=i}^{n}(\alpha_j-\alpha_{i-1})N_j(t)$$

$$s(t):=\sum_{i=0}^{n}\sigma_i u_i(t)\quad \textit{and}\quad \bar{s}(t):=\sum_{i=0}^{n}\sigma_i N_i(t),$$

and $(\sigma_i)_{i=0}^{n}$ is the sequence uniquely defined by

$$\sigma_0=\sigma_n=0,\quad \delta\sigma_i=-d_i,\ \textit{for all } i\in\{2,\ldots,n\}.$$

Proof. It can be checked that $\sum_{i=0}^{n}\alpha_i N_i(t)=t$ for all $t\in[a,b]$. The result follows from Theorem 1 if we prove that $v_i\ge\bar{v}_i$.

The curves $(t,v_i(t))^T$ and $(t,\bar{v}_i(t))^T$ can be seen as parametric curves generated by the systems $(u_0,\ldots,u_n)$ and $(N_0,\ldots,N_n)$, respectively, with the same control polygon $W_0\cdots W_n$ where

$$W_j:=\begin{cases}(\alpha_j,0)^T, & \text{if } j<i,\\ (\alpha_j,\alpha_j-\alpha_{i-1})^T, & \text{if } j\ge i.\end{cases}$$

The points $W_0,\dots,W_{i-1}$ are collinear and the points $W_{i-1},W_i,\dots,W_n$ are collinear too. Therefore all the control points $W_0,\dots,W_n$ lie on the polygon $W_0W_{i-1}W_n$. On the one hand, the parametric curve $(t,\bar{v}_i(t))^T$ describes the polygon $W_0W_{i-1}W_n$. On the other hand, by the convex hull property the curve $(t,v_i(t))^T$ is included in the inside of the triangle with vertices W_0, W_{i-1} and W_n . So we have that $v_i \geq \bar{v}_i$. □

Now let us apply the previous result to the particular case where $\gamma(t)$ is a Bernstein-Bézier curve and $\pi(t)$ is its corresponding control polygon. This result is not new. The bound was already obtained by Reif (Theorem 2.2 of [7]).

Corollary 3. *Let $(B_{0,n}(t),\dots,B_{n,n}(t))$ be the Bernstein basis of degree n defined on $[a,b]$ and let $\gamma(t)=\sum_{i=0}^n P_iB_{i,n}(t)$ and $\pi(t)$ the parameterization of its control polygon defined by (3). Then we have that*

$$\frac{1}{b-a}\int_a^b \|\gamma(t)-\pi(t)\|\,dt \leq \frac{n-1}{12}\max_{i\in\{2,\dots,n\}}\left\|\nabla^2 P_i\right\|.$$

Proof. We have that the Bernstein basis has the linear precision property

$$\sum_{i=0}^n \alpha_i B_{i,n}(t)=t, \quad \text{for all } t\in[a,b],$$

with $\alpha_i=a+i(b-a)/n$ for all $i\in\{0,\dots,n\}$. So we have that

$$\delta P_i=\frac{n}{b-a}\nabla^2 P_i, \quad i\in\{2,\dots,n\}.$$

Now we choose $d_i=2(b-a)/n$ for all $i\in\{2,\dots,n\}$ and check that the σ_i's introduced in Corollary 2 are of the form

$$\sigma_i=(\alpha_i-a)(b-\alpha_i)=\frac{i(n-i)}{n^2}(b-a)^2.$$

Then, by Corollary 2, we have

$$\int_a^b \|\gamma(t)-\pi(t)\|dt \leq \frac{n^2}{2(b-a)^2}\max\|\nabla^2 P_i\|\int_a^b D(t)dt,$$

and

$$\int_a^b D(t)=\sum_{i=1}^n \frac{(b-a)^2}{n^2}i(n-i)\int_a^b (B_{i,n}(t)-N_i(t))dt=$$
$$\frac{(b-a)^2}{n^2}\left(\frac{b-a}{n}-\frac{b-a}{n+1}\right)\sum_{i=1}^n i(n-i)=\frac{(b-a)^3}{n^3(n+1)}\frac{(n+1)n(n-1)}{6}$$

and the result follows. □

We observe that in Corollary 3

$$D(t):=\sum_{i=1}^{n-1}(\alpha_i-a)(b-\alpha_i)(N_{i,n}(t)-B_{i,n}(t))=(P-B)[(t-a)(b-t)],$$

where P is the operator transforming each function into its piecewise linear interpolant at $(\alpha_i, f(\alpha_i))$

$$P[f](t) := \sum_{i=0}^{n} f(\alpha_i) N_i(t), \quad t \in [a,b],$$

and B is the Bernstein operator

$$B[f](t) := \sum_{i=0}^{n} f(\alpha_i) B_{i,n}(t), \quad t \in [a,b].$$

In [1], the distance $\max_{t\in[a,b]} \|\gamma(t) - \pi(t)\|$ between a curve and its control polygon was analyzed. For the case of convexity preserving systems with the endpoint interpolation property, a practical bound was described in Proposition 2 of [1]. We restate below this result because it will be used for discussing bounds of the distance of rational curves to its control polygon.

Theorem 4. *Let $(u_0(t), \ldots, u_n(t))$ be a blending basis defined on $[a,b]$ such that there exists $\alpha_0 < \alpha_1 < \cdots < \alpha_n$ with $\sum_{i=0}^{n} \alpha_i u_i(t) = t$ and let $d_2, \ldots, d_n$ be a sequence of positive numbers. Assume that the representation is convexity preserving and that the endpoint interpolation property holds. Then, for any parametric curve $\gamma(t) = \sum_{i=0}^{n} P_i u_i(t)$ and the parameterization $\pi(t)$ of its corresponding control polygon given by (3), we have*

$$\max_{t\in[a,b]} \left\|\gamma(t) - \pi(t)\right\| \leq \max_{i\in\{2,\ldots,n\}} \left\| \frac{\delta P_i}{d_i} \right\| \max_{i\in\{1,\ldots,n-1\}} D(\alpha_i),$$

where $D(t)$ is defined by formula (6) .

3 Mean Distance from a Rational Curve to Its Control Polygon

Given a blending system of functions $(f_0, \ldots, f_n)$ defined on $[a,b]$ and some strictly positive weights $w_0, \ldots, w_n > 0$, a rational curve

$$\gamma(t) = \frac{1}{\sum_{i=0}^{n} w_i f_i(t)} \sum_{i=0}^{n} w_i f_i(t) P_i, \quad t \in [a,b], \tag{7}$$

can be defined. The rational curve can be interpreted as a curve defined by the blending system $(u_0, \ldots, u_n)$ with

$$u_i(t) := \frac{w_i f_i(t)}{\sum_{k=0}^{n} w_k f_k(t)}, \quad i = 0, \ldots, n. \tag{8}$$

Even in the simplest case of Bézier curves we cannot expect to have the linear precision property for all set of positive weights. If the linear precision property holds, then

$$t = \sum_{i=0}^{n} \alpha_i u_i(t) = \frac{\sum_{i=0}^{n} w_i \alpha_i f_i(t)}{\sum_{i=0}^{n} w_i f_i(t)}.$$

Therefore $t\sum_{i=0}^{n} w_i f_i(t)$ is a function in $\langle f_0,\dots,f_n\rangle$. For Bézier curves this implies that $\sum_{i=0}^{n} w_i t B_{i,n}(t)$ is a polynomial of degree less than $n-1$, and so

$$\Delta^n w_0 = \sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} w_i = 0.$$

In the case of B-spline curves this restriction on the weights is yet more complicated to express. For other spaces generated by blending systems of functions, the linear precision property would imply that all weights are equal reducing the problem to the nonrational one. For this reason, we want to relax the linear precision property.

Let us recall that a system $(f_0,\dots,f_n)$ is totally positive of order k (TP_k), if all the minors of its $k\times(n+1)$ collocation matrices

$$M\begin{pmatrix} f_0,\dots,f_n \\ t_0,\dots,t_{k-1}\end{pmatrix} := (f_j(t_i))_{0\le i\le k; 0\le j\le n}, \quad a\le t_0<t_1<\ldots<t_{k-1}\le b$$

are nonnegative. In the particular case that $k=n+1$, the system is called totally positive (TP).

In Theorem 2.6 of [2] it was shown that, if $(f_0,\dots,f_n)$ is a TP_2 blending system, then all functions

$$\frac{\sum_{i=0}^{n}\alpha_i w_i f_i(t)}{\sum_{i=0}^{n} w_i f_i(t)}, \quad w_0,\dots,w_n>0, \quad \alpha_0<\cdots<\alpha_n \tag{9}$$

are increasing. Moreover, if for some particular choice of w_i and α_i, $i=0,\dots,n$, a function (9) is strictly increasing, then all functions (9) are also strictly increasing for any choice of w_i and α_i, $i=0,\dots,n$.

In Corollary 4.6 of [2] it was shown that, if $(f_0,\dots,f_n)$ is a TP_3 blending system and for some particular choice of w_i and α_i, $i=0,\dots,n$, the function (9) is strictly increasing, then for any choice of the weights w_i and α_i, $i=0,\dots,n$, the blending system $(u_0\circ\alpha^{-1},\dots,u_n\circ\alpha^{-1})$ is convexity preserving, where $u_i(t)$ is defined by (8) and

$$\alpha(t) := \frac{\sum_{i=0}^{n}\alpha_i w_i f_i(t)}{\sum_{i=0}^{n} w_i f_i(t)} = \sum_{i=0}^{n}\alpha_i u_i(t). \tag{10}$$

We also observe that $\sum_{i=0}^{n}\alpha_i u_i\circ\alpha^{-1}(t)=t$, and we recover the linear precision property for the rational system $(u_0\circ\alpha^{-1},\dots,u_n\circ\alpha^{-1})$.

Assume that for some strictly increasing sequence $a=\alpha_0<\cdots<\alpha_n=b$ both functions

$$\alpha(t) := \sum_{i=0}^{n}\alpha_i u_i(t), \quad \bar\alpha(t) := \sum_{i=0}^{n}\alpha_i \bar u_i(t)$$

are strictly increasing. This condition automatically holds for non-rational and for rational Bézier and B-spline representations with strictly positive weights because they are totally positive bases. In Section 4 of [4], we can find more examples of systems of functions satisfying this condition. If the endpoint interpolation property holds, then $\alpha(a)=a$ and $\alpha(b)=b$ and we deduce that $\alpha([a,b])=[a,b]=\bar\alpha([a,b])$.

The main idea is that we do not compare points in both curves with the same parameter value. The distance between both curves is measured comparing corresponding

points of both curves $\gamma\circ\alpha^{-1}$ and $\bar\gamma\circ\bar\alpha^{-1}$. A suitable choice of α_i (for instance, equidistant values in $[a,b]$) might give rise to parameter changes leading to reasonable distance bounds. Let us observe first that $(u_0\circ\alpha^{-1},\dots,u_n\circ\alpha^{-1})$ and $(\bar u_0\circ\bar\alpha^{-1},\dots\bar u_n\circ\bar\alpha^{-1})$ are blending systems with the linear precision property and we can apply to them the results of the previous section. From Theorem 1 we have

$$\int_a^b \left\|\gamma(\alpha^{-1}(t))-\bar\gamma(\bar\alpha^{-1}(t))\right\| w(t)dt \le \max_{i\in\{2,\dots,n\}}\frac{\|\delta P_i\|}{d_i}\int_a^b D(t)w(t)dt$$

where $D(t):=\sum_{i=2}^n d_i\,|v_i(t)-\bar v_i(t)|$ and

$$v_i(t):=\sum_{j=i}^n(\alpha_j-\alpha_{i-1})u_j(\alpha^{-1}(t)),\quad \bar v_i(t):=\sum_{j=i}^n(\alpha_j-\alpha_{i-1})\bar u_j(\bar\alpha^{-1}(t)).$$

The change of variables $t=\alpha(s)$ gives rise to

$$\int_a^b D(t)w(t)dt=\int_a^b D(\alpha(s))w(\alpha(s))\alpha'(s)ds,$$

which might have a simpler form. However, for the effective computation of the above bound and other estimates, it might be required to invert the function α. Since α is strictly increasing, the solution s of the equation $\alpha(s)-t=0$ can be obtained by the bisection method. We have tested Newton's method in the particular case of rational Bézier curves with fast and accurate results. In this way, we have computed $t_i=\alpha^{-1}(\alpha_i)$ in bound (11) of Theorem 3 for the numerical tests at the end of this section.

Observe that if $\bar\gamma$ is the representation of the control polygon (3), then $\bar\alpha(t)=t$ and Corollary 2 gives

$$\int_a^b \left\|\gamma(\alpha^{-1}(t))-\pi(t)\right\| dt \le \max_{i\in\{2,\dots,n\}}\frac{\|\delta P_i\|}{d_i}\int_a^b D(t)\,dt$$

with $D(t)=\sum_{i=2}^n d_i(v_i(t)-\bar v_i(t))=\bar s(t)-s(t)$,

$$s(t):=\sum_{i=0}^n\sigma_i u_i(\alpha^{-1}(t)),\quad \bar s(t):=\sum_{i=0}^n\sigma_i N_i(t),$$

and $(\sigma_i)_{i=0}^n$ the sequence defined by

$$\sigma_0=\sigma_n=0,\quad \delta\sigma_i=-d_i,\ \text{for all } i\in\{2,\dots,n\}.$$

Theorem 4 can also be used for analyzing the distance between a rational curve and its control polygon.

Theorem 5. *Let $(f_0,\dots,f_n)$ be a TP_3 blending system satisfying the endpoint interpolation property. Let $a=\alpha_0<\alpha_1<\cdots<\alpha_n=b$ be such that $\alpha(t)=\sum_{i=0}^n\alpha_i u_i(t)$ is an strictly increasing function, with the functions $u_0,u_1,\dots,u_n$ given by (8), and let $d_2,\dots,d_n$ be a sequence of positive numbers. Then, for any set of weights $w_0,\dots,w_n>0$ the curve*

$$\gamma(t)=\frac{1}{\sum_{i=0}^n w_i f_i(t)}\sum_{i=0}^n w_i f_i(t)P_i,$$

satisfies

$$\|\gamma(\alpha^{-1}(t)) - \pi(t)\| \leq \max_{i\in\{2,\dots,n\}} \frac{\|\delta P_i\|}{d_i} \max_{i=1,\dots,n-1} D(\alpha_i) \tag{11}$$

where

$$D(\alpha_i) = \sigma_i - \frac{\sum_{k=0}^n w_k \sigma_k f_k(t_i)}{\sum_{k=0}^n w_k f_k(t_i)}$$

and $t_i := \alpha^{-1}(\alpha_i)$.

Proof. Let us define $u_i(t)$, $i = 0,\dots,n$ by (8). Then $(u_0 \circ \alpha^{-1}, \dots, u_n \circ \alpha^{-1})$ satisfies the linear precision property by (10) and is convexity preserving by Corollary 4.6 of [2]. Observe that since $(f_0,\dots,f_n)$ satisfies the endpoint interpolation property, then we have that

$$u_i(a) = \frac{w_i f_i(a)}{w_0 f_0(a)} = \delta_{i0}, \quad u_i(b) = \frac{w_i f_i(b)}{w_n f_n(b)} = \delta_{in}, \quad i = 0,\dots,n,$$

and, since $\alpha(a) = a$, $\alpha(b) = b$, we deduce that $(u_0 \circ \alpha^{-1}, \dots, u_n \circ \alpha^{-1})$ also satisfies the endpoint interpolation property. Applying Theorem 4 to the curve $\gamma(\alpha^{-1}(t)) = \sum_{i=0}^n u_i \circ \alpha^{-1}(t) P_i$, we obtain (11), where

$$D(t) = \sum_{k=0}^n \sigma_k (N_k(t) - u_k \circ \alpha^{-1}(t)) = \sum_{k=0}^n \sigma_k N_k(t) - \frac{\sum_{k=0}^n \sigma_k w_k f_k(\alpha^{-1}(t))}{\sum_{k=0}^n w_k f_k(\alpha^{-1}(t))}.$$

By (3), we have that $\sum_{k=0}^n \sigma_k N_k(\alpha_i) = \sigma_i$ and defining $t_i := \alpha^{-1}(\alpha_i)$ we may write

$$D(\alpha_i) = \sigma_i - \frac{\sum_{k=0}^n \sigma_k w_k f_k(t_i)}{\sum_{k=0}^n w_k f_k(t_i)}.$$

□

We observe that the values $D(\alpha_i)$ depend on the weights $w_0,\dots,w_n$. A question which arises in a natural way is the choice of the weights $w_0,\dots,w_n$ such that the bound in (11) is the least possible. We will study the important particular case where $(f_0,\dots,f_n)$ is the Bernstein basis:

$$f_i(t) = \binom{n}{i} t^i (1-t)^{n-i}. \quad i = 0,1,\dots,n.$$

We have chosen $\alpha_i = i/n$, $i = 0,1,\dots,n$, and $d_i = \alpha_i - \alpha_{i-2}$, $i = 2,\dots,n$. A first attempt revealed that the determination of an optimal choice of the weights has a complicated analysis. Therefore, we tried to determine numerically almost optimal choices and suggest a choice of the weights that may give near optimal results, at least for low degrees.

For our experiments we used degrees $n = 5$, $n = 8$ and $n = 10$ and we obtained almost optimal and symmetrical choices of the weight vector,

$$\mathbf{w}_5^{aopt} = (1,4,12,12,4,1).$$
$$\mathbf{w}_8^{aopt} = (1,8.5,110,400,700,400,110,8.5,1).$$
$$\mathbf{w}_{10}^{aopt} = (1,15.6,452,3500,14764,22120,14764,3500,452,15.6,1).$$

Table 1. Distance measures for different weights and degrees

n	weights	$\max_{0\le i\le n} D(\alpha_i)$
5	$\mathbf{w}^{unit}$	4.8000×10^{-2}
	$\mathbf{w}^{aopt}$	2.3732×10^{-2}
	$\mathbf{w}^{nopt}$	2.5687×10^{-2}
8	$\mathbf{w}^{unit}$	3.1250×10^{-2}
	$\mathbf{w}^{aopt}$	1.1516×10^{-2}
	$\mathbf{w}^{nopt}$	1.2279×10^{-2}
10	$\mathbf{w}^{unit}$	2.5000×10^{-2}
	$\mathbf{w}^{aopt}$	7.8476×10^{-3}
	$\mathbf{w}^{nopt}$	8.4429×10^{-3}

We also found that this values were similar to those obtained by the formula

$$\mathbf{w}_n^{nopt} = ((3/2)^{i(n-i)}, i = 0, \ldots, n)$$

and checked that this choice lead to near optimal results, at least for low degrees. In Table 1, we can find the value of $\max D(\alpha_i)$, for different degrees and different choices of the weight vector. We observe that the distance can be reduced even by a factor greater than 3 in some cases.

We have also analyzed the behaviour in the case of degrees much higher than 10. We have obtained significative reduction of the distance but with the disadvantage of using very high weights. The values of the $t_i = \alpha^{-1}(\alpha_i)$ tend to concentrate to the ends of

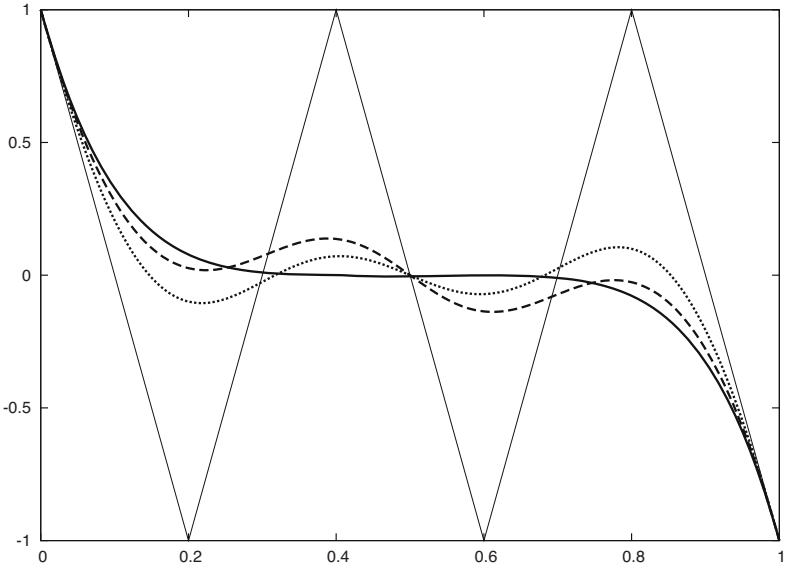


Fig. 2. Rational curves with different choices of weights

the interval, leading to curves with very high change rate and high values of $\gamma'(t)$ at the ends of the curve. For this reason, we only recommend using near optimal weights for moderate degrees up to 12.

In Figure 2, we have depicted three rational curves of degree 5 corresponding to the control polygon $P_0 \cdots P_5$, with $P_i = (i/5, (-1)^i)$, $i = 0, 1, \ldots, 5$. The solid line corresponds to unit weights, that is, the Bézier curve. The dashed line corresponds to the choice $\mathbf{w}_5^{nopt}$, whereas the dotted line corresponds to $\mathbf{w}_5^{aopt}$. In both cases we found that the curves imitate better the shape of the control polygon than in the Bézier case.

References

1. Carnicer, J.M., Floater, M.S., Peña, J.M.: The distance of a curve to its control polygon. Numerical methods of approximation theory and computer aided geometric design. RACSAM Rev. R. Acad. Cienc. Exactas Fís. Nat. Ser. A Mat. 96, 175–183 (2002)
2. Carnicer, J.M., García-Esnaola, M., Peña, J.M.: Convexity of rational curves and total positivity. J. Comput. Appl. Math. 71, 365–382 (1996)
3. Carnicer, J.M., Peña, J.M.: Monotonicity preserving representations. In: Laurent, P.J., Le Méhauté, A., Schumaker, L.L. (eds.) Curves and Surfaces in Geometric Design, pp. 83–90. AKPeters, Wellesley (1994)
4. Delgado, J., Peña, J.M.: Corner cutting systems. Computer Aided Geometric Design 22, 81–97 (2005)
5. Karavelas, M.I., Kaklis, P.D., Kostas, K.V.: Bounding the Distance between 2D Parametric Bézier Curves and their Control Polygon. Computing 72, 117–128 (2004)
6. Nairn, D., Peters, J., Lutterkort, D.: Sharp, quantitative bounds on the distance between a polynomial piece and its Bézier control polygon. Computer Aided Geometric Design 16, 613–631 (1999)
7. Reif, U.: Best bound on the approximation of polynomials and splines by their control structure. Computer Aided Geometric Design 17, 579–589 (2000)

Compactly Supported Splines with Tension Properties on a Three-Direction Mesh

Paolo Costantini[1], Francesca Pelosi[2], and Maria Lucia Sampoli[1]

[1] Dip. di Scienze Matematiche ed Informatiche, Università di Siena
Pian dei Mantellini 44, Siena, Italy
[2] Dipartimento di Matematica, Università di Roma "Tor Vergata",
Via della Ricerca Scientifica 1, Roma, Italy

Abstract. In this paper we show that for a regular triangulation it is possible to construct composite triangular C^1 functions with local tension properties which are non-negative and compactly supported. Moreover, they form a partition of unity and reproduce first degree polynomials.

1 Introduction

Piecewise functions with tension properties are nowadays very popular and are profitably used in several practical problems, ranging from the construction of curves and surfaces in free form design to shape preserving interpolation or approximation. The univariate cases and their tensor product counterparts are relatively simple and well studied. In particular, in some cases B-spline like basis with tension properties have been explicitly computed (see, as typical examples, those reported in [7]) and, among the general results we recall that any *spline space* composed by piecewise functions taken from Chebyshev spaces admits a B-spline basis [3,18,21].

The construction of piecewise bivariate functions with tension properties defined over a triangulation is far more complicated; however, some triangular macro-elements, defined on Clough-Tocher or Powell-Sabin splits, [8,9,10], and single triangular elements with tension properties, [13], have been recently proposed and used in the construction of composite C^1 functions. While for the classical Powell-Sabin quadratic macro-elements an effective basis consisting of positive compactly supported functions have been proposed, [14], similar results have not been obtained for functions with tension properties.

In this paper we present a first result in this direction. We start by defining a class of variable degree polynomial triangular elements, where the independent degrees, k_1, k_2, k_3, associated with the vertices, P_1, P_2, P_3, of the triangular domain play the role of tension parameters. These elements are closely related to those described in [13] and tend to affine functions for large degrees. They belong to a nine dimensional subspace of $\mathbb{P}_{\max\{k_1,k_2,k_3\}}$ isomorphic to the finite element space of polynomials of total degree five. Their normal derivatives across the edges are computed as a function of the normal derivatives at vertices and their second derivatives vanish at the vertices.

Then we construct a partition of unity composed of non-negative C^1 piecewise polynomial functions with hexagonal support centered at the vertices of a uniform equilateral

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 93–110, 2010.

triangulation of the plane, and generating a space containing first degree polynomials. These functions are C^2 –with vanishing second derivatives– at the vertices of the triangulation, partially resembling *super splines* (see, e.g. [5,22]). It is worth noticing that, since we work with spaces isomorphic to subspaces of quintic polynomials, C^2 functions could in principle be obtained (see [17,19]); here the additional available parameters are used to fulfill the tension properties. It is also worth anticipating that their support consists of 24 triangles and that there are 12 non-vanishing basis functions on each triangle. Since the dimension of the triangular finite element space is 9, we have a redundant spanning system. This fact is common in bivariate triangular splines, as pointed out, e.g., in [1,4,6,20].

The paper is divided into five sections. In the next one we recall some univariate results about the variable degree spaces which constitute the building blocks for the construction of the triangular element. In Sect. 3 we describe the nine parameter triangular elements with tension properties; more precisely, we show that they can be defined by assigning Hermite conditions at the vertices of the triangle and that it is possible to use them for locally constructing composite C^1 surfaces on general triangulations. It is worth mentioning that also the degrees are local, that is are independently assigned to each vertex, so that the surface assumes a planar or cuspidal behavior only where needed. Then in Sect. 4 we consider composite surfaces on uniform triangulations and show that it is possible to construct a set of functions with the above mentioned properties. Section 5 is devoted to final comments and examples of applications.

2 Background Material

In order to make the paper self-readable, we start by presenting some preliminary material. Let us briefly recall the main definitions and properties of *quintic-like* variable degree polynomials, referring to [11] and [12] for full details.

Given a (vector) parameter $\mathbf{k} = (k_0, k_1)$ with $k_i \geq 5$ and $t \in [0,1]$[1], we consider the six-dimensional, variable degree polynomial space

$$VP_5^{\mathbf{k}} = \text{span}\left\{(1-t)^{k_0}, t(1-t)^{k_0-1}, \mathbb{P}_1, (1-t)t^{k_1-1}, t^{k_1}\right\} \quad . \tag{1}$$

Clearly $VP_5^{\mathbf{k}}$ is isomorphic to $\mathbb{P}_5$ and in particular we have $VP_5^{(5,5)} = \mathbb{P}_5$. Its main property is the geometric control of its elements. Indeed, it is shown in [11] that $VP_5^{\mathbf{k}}$ admits a positive, normalized *pseudo-Bernstein basis*, $\{\mathscr{B}_j^{\mathbf{k}}\}$. Therefore, any $\phi \in VP_5^{\mathbf{k}}$ can be expressed as $\phi = \sum_{j=0}^5 b_j \mathscr{B}_j^{\mathbf{k}}$, and its shape can be determined by the *pseudo control polygon* $\ell_\phi = \ell_\phi(t)$, that is by the piecewise linear function such that

$$\ell_\phi(\xi_j) = b_j \; ; \; j = 0, 1, \ldots, 5 \quad , \tag{2}$$

where the *Greville abscissas* are given by

$$\xi_0 = 0, \;\; \xi_1 = \frac{1}{k_0}, \;\; \xi_2 = \frac{2}{k_0}, \;\; \xi_3 = 1 - \frac{2}{k_1}, \;\; \xi_4 = 1 - \frac{1}{k_1}, \;\; \xi_5 = 1 \quad .$$

[1] We use the interval $[0,1]$ for notational simplicity; the extension to general intervals is straightforward.

The pairs $(\xi_0, b_0), \ldots, (\xi_5, b_5)$ are called *pseudo-Bézier control points*. Obviously $VP_5^{\mathbf{k}} \subseteq \mathbb{P}_{\tilde{k}}$, $\tilde{k} = \max\{k_0, k_1\}$, and therefore any $\phi \in VP_5^{\mathbf{k}}$ can be represented also using the (redundant) $\tilde{k}$-degree Bernstein basis $\{B_0^{(\tilde{k})}, B_1^{(\tilde{k})}, \ldots, B_{\tilde{k}}^{(\tilde{k})}\}$ and described by the corresponding $\tilde{k}$-degree Bézier net $\lambda_\phi = \lambda_\phi(t)$.

An important property of $VP_5^{\mathbf{k}}$ (see [11], Subsection 3.1) makes easier both the practical control and the theoretical analysis of the shape of variable degree curves. It can be stated as follows.

Theorem 1. *Let $\phi \in VP_5^{\mathbf{k}}$, let ℓ_ϕ be defined by (2) and let λ_ϕ be the $\tilde{k}$-degree Beziér net of ϕ. Then there exists a corner cutting operator $\mathscr{D}_{\mathbf{k}}^{\tilde{k}}$ such that*

$$\lambda_\phi = \mathscr{D}_{\mathbf{k}}^{\tilde{k}} \ell_\phi \quad . \tag{3}$$

The operator $\mathscr{D}$ can be viewed as a *one sided degree-elevation* as shown in Fig. 1 for the cases $k_0 = 5, k_1 = 7$ and $k_0 = 7, k_1 = 5$. From the well-known results of [16] we know that sufficient conditions on the shape of a Bézier curve can be inferred from the (discrete) shape of its control polygon and, from the above property, the shape of the control polygon reproduces that of the *pseudo control polygon*. Therefore, we can control the shape of a variable degree curve using the mutual position of only six points, independently of the degrees k_0 and k_1.

Let L_q denote the space of continuous piecewise linear functions with break points i/q, $i = 0, \ldots, q$. Let $n, m \in \mathbb{N}$, $m \leq n$, and let $\mathscr{E}_m^n$ the *degree raising operator*

$$\begin{gathered} \mathscr{E}_m^n \ : \ L_m \mapsto L_n \ \text{ s.t. } \forall\, f \in L_m \quad , \\ \sum_{i=0}^{m} f\left(\frac{i}{m}\right) B_i^{(m)} = \sum_{i=0}^{n} (\mathscr{E}_m^n f)\left(\frac{i}{n}\right) B_i^{(n)} \quad . \end{gathered} \tag{4}$$

As mentioned in the introduction, the degrees k_0, k_1 act as tension parameters for the spaces (1): indeed, from the properties of the involved functions we immediately have the following asymptotic result (see also the similar results in [11] for *spline* spaces).

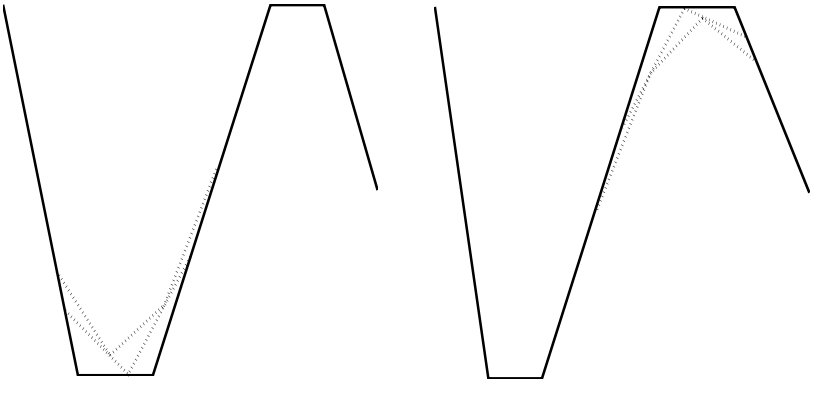


Fig. 1. Examples of the operator $\mathscr{D}_{\mathbf{k}}^{\tilde{k}}$

Theorem 2. *Let* $\phi^{\mathbf{k}} \in VP_5^{\mathbf{k}}$ *be such that*

$$\frac{d^q}{dt^q}\phi^{\mathbf{k}}(p) = f_p^q \ ; \ p = 0,1 \ ; \ q = 0,1,2 \ \ ,$$

for given values f_p^q*. Then, setting* $\underline{k} := \min\{k_0, k_1\}$*, for any* $t \in [a,b] \subset (0,1)$

$$\lim_{\underline{k} \to +\infty} \frac{d^2}{dt^2}\phi^{\mathbf{k}}(t) = 0 \ \ .$$

3 The Triangular Element

In this section we construct a triangular element with tension properties. This element, closely related to and simpler than that described in [13], is particularly suited for the compactly supported functions of Sect. 4.

Given a triangular domain, we want to construct a triangular polynomial element depending on shape parameters and possessing tension properties, that is tending to an affine function for limit values of the shape parameters. As usual we shall describe the triangular surface patch in terms of its Bernstein-Bézier net (see e.g. [15]). Let $T \subset \mathbb{R}^2$ be a non-degenerate triangle with vertices P_1, P_2, P_3. The barycentric coordinates u_1, u_2, u_3 are such that any point $P \in T$ can be expressed as

$$P = \sum_{i=1}^{3} u_i P_i,$$

where, setting $u = (u_1, u_2, u_3)$ and using standard multi-index notation, we impose $u_i \geq 0$ and $|u| = 1$. A bivariate n-degree Bernstein-Bézier polynomial can be written in the form

$$p(u) = \sum_{|\alpha|=n} b_\alpha^{(n)} B_\alpha^{(n)}(u) \text{ where } B_\alpha^{(n)}(u) = \frac{n!}{\alpha!} u^\alpha \ \ , \tag{5}$$

for $\alpha = (\alpha_1, \alpha_2, \alpha_3)$, with $|\alpha| = n$ and using classical conventions for $u^\alpha = u_1^{\alpha_1} u_2^{\alpha_2} u_3^{\alpha_3}$ and $\alpha! = \alpha_1! \alpha_2! \alpha_3!$. The functions $B_\alpha^{(n)}(u)$ are called *triangular Bernstein polynomials* and the points $b_\alpha^{(n)}$ are the corresponding Bernstein-Bézier coefficients.

As in the univariate case we may associate to each B-B-polynomial a Bézier control net whose values are

$$b_\alpha^{(n)} = \left(\frac{\alpha}{n}, b_\alpha^{(n)} \right), \ |\alpha| = n \ \ .$$

For simplicity we shall refer to $b_\alpha^{(n)}$ as control points as well.

The basic idea behind the construction of the triangular element is to associate to each triangle vertex a variable degree $k_i \geq 5$, $i = 1,2,3$ and given $\overline{k} = \max\{k_1, k_2, k_3\}$, to define a space $VP_5^{\triangle,\overline{k}}$ such that

$$VP_5^{\triangle,\overline{k}} \subset \mathbb{P}_{\overline{k}} \ ; \ \dim VP_5^{\triangle,\overline{k}} = 9 \ \ . \tag{6}$$

In the Bernstein-Bézier representation each element of $\mathbb{P}_{\overline{k}}$ will be uniquely determined once the $(\overline{k}+2)(\overline{k}+1)/2$ Bézier control points are assigned. The construction of a subspace of dimension 9 will imply that, given 9 free parameters, it is possible to determine all the control points.

We shall see in the following that once 9 conditions (3 for each vertex) are given, all the control points are obtained by imposing linear relations/shape constraints with a similar approach to that used in the construction of the univariate space (1).

As usual in the Finite Element literature, the triangular element is defined by Hermite interpolation conditions at each vertex: function and gradient values,

$$f(P_r)\ ;\ f_x(P_r), f_y(P_r)\ ;\ r=1,2,3\ \ . \tag{7}$$

The construction can be sketched as follows: the Hermite conditions are used to determine the 3 control points around the vertices on 3 polynomials of degree k_1, k_2 and k_3 respectively:

$$\{b^{(k_i)}_{k_i e_i}, b^{(k_i)}_{(k_i-1)e_i+e_j}, j\neq i\}\ ,\ i=1,2,3\ \ . \tag{8}$$

If we impose now vanishing second derivatives at each vertex, we have that the next three control points around the vertices, i.e. $b^{(k_i)}_{\alpha}$ with $\alpha_i = k_i - 2$, lie on the same plane. For instance the control points $b^{(k_1)}_{\alpha}$ with $\alpha_1 = k_1 - 2$, i.e., the control points

$$\{b^{(k_1)}_{(k_1-2)e_1+2e_2}, b^{(k_1)}_{(k_1-2)e_1+e_2+e_3}, b^{(k_1)}_{(k_1-2)e_1+2e_3}\},$$

lie on the plane defined by $\{b^{(k_1)}_{k_1e_1}, b^{(k_1)}_{(k_1-1)e_1+e_2}, b^{(k_1)}_{(k_1-1)e_1+e_3}\}$, and can be obtained using straightforward computations (see [15]).

Having obtained (8), we can completely determine all the control points of a $\overline{k}$ degree Bézier net. The construction is done in three steps.

First we construct the control points along the edges P_iP_{i+1}, where $i-1\in\mathbb{Z}_3$, i.e the sum is performed $\mathtt{mod}_3$,

$$b^{(\overline{k})}_{(\overline{k}-r)e_i+re_{i+1}},\quad r=0,\ldots,\overline{k}\ ; \tag{9}$$

then we determine the control points along the second lines parallel to the edges, and finally along al the remaining parallel lines.

Let us consider now in detail the construction of the edge control points. We may describe the construction of the edge P_1P_2 as a similar procedure can be used for the other ones. We recall that the sections of a bivariate $\overline{k}$-degree polynomial of the form (5) along isoparametric lines, for instance $u_3 = \text{const}$, are univariate $\overline{k}$-degree polynomials defined in the interval $[0, 1-u_3]$.

The Hermite conditions relative to the present edge imply to compute a function $\psi = \psi(t)\ ,\ t\in[0,1]\ ,\ \psi\in VP^{\mathbf{k}}_5$, with $\mathbf{k}=(k_1,k_2)$, such that

$$\begin{aligned}&\psi(0)=f(P_1),\ \psi'(0)=f_t(P_1),\ \psi''(0)=f_{tt}(P_1)=0\ ;\\ &\psi(1)=f(P_2),\ \psi'(1)=f_t(P_2)\ \ ,\ \psi''(1)=f_{tt}(P_2)=0\ ;\end{aligned}$$

Setting $\widetilde{k} = \max\{k_1, k_2\}$ and recalling the definition of Bézier control nets (2) and (3), the control points of the considered edge can be obtained from the degree elevated net of ψ:

$$b^{(\overline{k})}_{(\overline{k}-r)e_1+re_2} = \lambda_\psi\left(\frac{r}{\overline{k}}\right),\ r=0,\dots,\overline{k};\ \ \lambda_\psi = \mathscr{E}^{\overline{k}}_{\widetilde{k}}\mathscr{D}^{\widetilde{k}}_{\mathbf{k}}\ell_\psi\,. \tag{10}$$

Let us move now to the construction of the second lines. As before the corresponding control points

$$b^{(\overline{k})}_{\alpha},\ \text{with}\ \ \alpha_i = 1,\ i=1,2,3, \tag{11}$$

are seen as the control points of a degree elevated control polygon of a univariate variable degree polynomial. In more detail, considering for instance the line $u_3 = 1/\overline{k}$, parallel to P_1P_2, the extreme points values

$$b^{(\overline{k})}_{(\overline{k}-1)e_1+e_3}, b^{(\overline{k})}_{(\overline{k}-2)e_1+e_2+e_3}, b^{(\overline{k})}_{e_1+(\overline{k}-2)e_2+e_3}, b^{(\overline{k})}_{(\overline{k}-1)e_2+e_3}\ ,$$

can be seen as given by the Hermite conditions (7). The other control points are computed as a function of those ones.

In order to explain the idea more thoroughly, let us introduce the following notation. Let $n \in \mathbb{N},\ n \geq 5$ and denote with $\pi^{(n)}_\alpha$ the plane such that

$$\pi^{(n)}_\alpha\left(\frac{\beta}{n}\right) = b^{(n)}_\beta, \qquad \beta \in \{\alpha, \alpha - e_1 + e_2, \alpha - e_1 + e_3\},\ \alpha_1 \neq 0$$

and with δ^n_α the cross boundary normal derivative (n.d. for short) of $\pi^{(n)}_\alpha$ with respect to the edge $b^{(n)}_\alpha b^{(n)}_{\alpha-e_1+e_2}$. We have that, calling H the orthogonal projection of P_3 into the line P_1P_2, and with ρ_H the corresponding parameter value (i.e. $H = (1-\rho_H)P_1 + \rho_H P_2$)),

$$\delta^{(n)}_\alpha = \frac{n}{\|P_3 - H\|}\left[b^{(n)}_{\alpha-e_1+e_3} - \left((1-\rho_H)b^{(n)}_\alpha + \rho_H b^{(n)}_{\alpha-e_1+e_2}\right)\right]\ .$$

Let us consider now $n = k_1$ (for P_1) and $n = k_2$ (for P_2). Due to the vanishing second derivatives at the vertices, we have the n.d. across the first two points and the last two are equal:

$$\delta^{(k_1)}_{k_1e_1} = \delta^{(k_1)}_{(k_1-1)e_1+e_2}\ ;\ \delta^{(k_2)}_{2e_1+(k_2-2)e_2} = \delta^{(k_2)}_{e_1+(k_2-1)e_2}\ .$$

Let $\delta = \delta(t)\,, t \in [0,1]$ be the piecewise linear function connecting the points

$$\left(0, \delta^{(k_1)}_{k_1e_1}\right),\ \left(\frac{1}{k_1}, \delta^{(k_1)}_{k_1e_1}\right)\ ,\left(\frac{2}{k_1}, \delta^{(k_1)}_{k_1e_1}\right), \left(\left(1-\frac{2}{k_2}\right), \delta^{(k_2)}_{e_1+(k_2-1)e_2}\right)\ ,$$

$$\left(\left(1-\frac{1}{k_2}\right), \delta^{(k_2)}_{e_1+(k_2-1)e_2}\right),\ \left(1, \delta^{(k_2)}_{e_1+(k_2-1)e_2}\right)\ .$$

We point out that δ is the pseudo-Bézier control polygon of a variable degree polynomial in $VP^{\mathbf{k}}_5$ where now $\mathbf{k} = \{k_1, k_2\}$ and we construct the degree elevated piecewise linear function

$$d = d(t) := \mathscr{E}^{\overline{k}}_{\widetilde{k}}\mathscr{D}^{\widetilde{k}}_{\mathbf{k}}\delta\ .$$

Finally, recalling the definition of λ_ψ in (10), we set

$$b_\alpha^{(\overline{k})} := \lambda_\psi(\xi_s) + d(\xi_s)\frac{\|P_3 - H\|}{\overline{k}},\ \alpha_3 = 1,\ \alpha_2 = s\ ,\ s = 0,\ldots,\overline{k}-1\ \ , \tag{12}$$

where $\xi_s = (s+\rho_H)/\overline{k}$. We can repeat a similar construction for the edges P_2P_3 and P_3P_1, defining all the control points (11).

Now, if $\overline{k} = 5$ all the control points are already determined. Let us describe the construction for the remaining lines, for $\overline{k} \geq 6$, in order to determine the points $b_\alpha^{(\overline{k})}$ with $2 \leq \alpha_i \leq \overline{k}-4$. We use three preliminary sets of internal control points, namely $b1_\alpha^{(\overline{k})}$, $b2_\alpha^{(\overline{k})}$ and $b3_\alpha^{(\overline{k})}$. The first one is constructed using lines parallel to the edge P_2P_3, that is, for any $\alpha_1 = 2,\ldots,\overline{k}-4$, $b1_\alpha^{(\overline{k})}$ is on the straight line connecting the (already computed) control points $b^{(\overline{k})}_{\alpha_1 e_1 + (\overline{k}-1-\alpha_1)e_2+e_3}$ and $b^{(\overline{k})}_{(\overline{k}-1-\alpha_1)e_1+\alpha_1 e_2+e_3}$. Similarly, the second and third set are computed using lines parallel to the edges P_3P_1 and P_1P_2 respectively. Finally, the internal control points are computed by a weighted sum:

$$b_\alpha^{(\overline{k})} = \left(\frac{\alpha_1-2}{\overline{k}-6}b1_\alpha^{(\overline{k})} + \frac{\alpha_2-2}{\overline{k}-6}b2_\alpha^{(\overline{k})} + \frac{\alpha_3-2}{\overline{k}-6}b3_\alpha^{(\overline{k})}\right)\ \ . \tag{13}$$

We remark that for $\overline{k} = 6$ we have to determine a unique internal point. Each set $b1_\alpha^{(\overline{k})}$, $b2_\alpha^{(\overline{k})}$ and $b3_\alpha^{(\overline{k})}$ reduces to a single point, then the internal point can be simply computed as a mean of them. Note also that in (13) the weights are chosen so that the three values $b1_\alpha^{(\overline{k})}$, $\alpha_1 = \overline{k}-4$; $b2_\alpha^{(\overline{k})}$, $\alpha_2 = \overline{k}-4$; $b3_\alpha^{(\overline{k})}$, $\alpha_3 = \overline{k}-4$ are exactly interpolated.

Of course the proposed choice (13) is not the only possible one; for instance the inner control points could have been placed directly on a plane connecting the three points specified above, as in Fig. 2(**d**). On the other hand, we may observe that formulas (10) and (12) both reproduce the good features of (1) and facilitate the C^1 continuity between adjacent patches (see the next section), while formulas (13) provides nice shapes when we have different degrees (the shapes of the second lines, different around the vertices, are smoothly blended when moving toward the center).

Remark 1. *If the data (7) are taken from an affine function, (10), (12) and (13) imply that all the control points $b_\alpha^{(\overline{k})}$ lie on the same affine function.*

Using Theorem 2, the construction defined by (13) and the properties of derivatives of triangular Bernstein polynomials ([15]), we have the following result.

Theorem 3. *Let the data (7) be given, let $VP_5^{\triangle,\overline{k}}$ be given by (6), and let $\Phi^{(\overline{k})} \in VP_5^{\triangle,\overline{k}}$. Let $C \subset \{T \setminus \{P_1P_2, P_2P_3, P_3P_1\}\}$ be any compact set. Then, for any $(x,y) \in C$*

$$\lim_{k_1,k_2,k_3\to\infty} \frac{\partial^2}{\partial x^\mu \partial y^\nu}\Phi^{(\overline{k})}(x,y) = 0, \quad \mu+\nu = 2\ \ .$$

Essentially, Theorem 3 says that the functions of the space (6) have *tension properties* because the sequence $\Phi^{(\overline{k})}$ tends uniformly, in any compact C, to the plane connecting the points $(P_r, f(P_r))$, $r = 1,2,3$. Moreover, the convergence is *smooth* since the

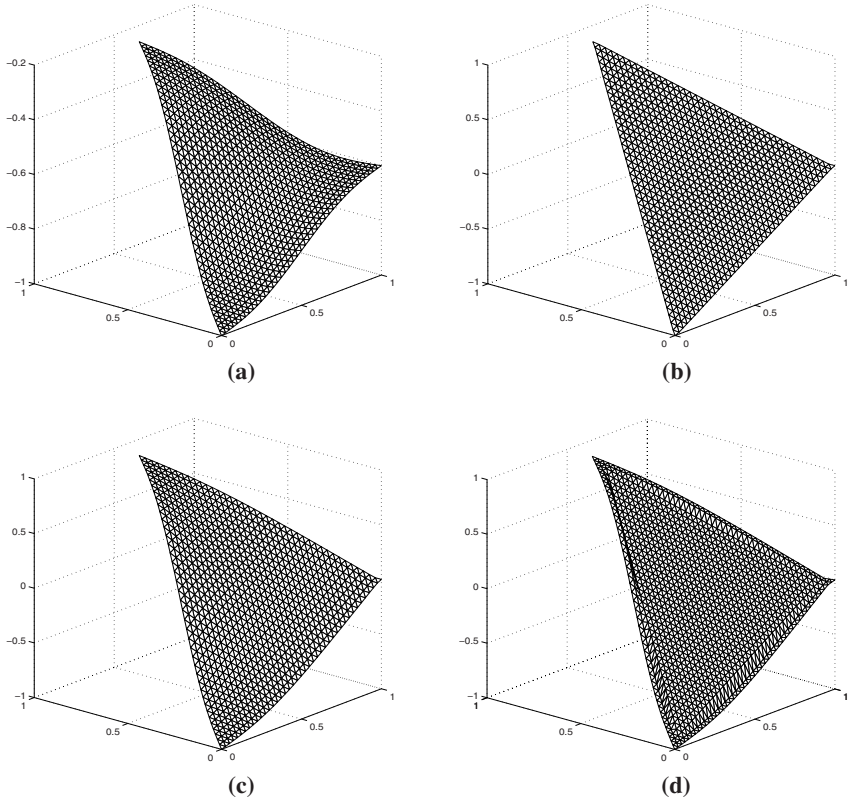


Fig. 2. The triangular element: **(a)** $k_1 = k_2 = k_3 = 5$; **(b)** $k_1 = k_2 = k_3 = 100$; **(c)**-**(d)** $k_1 = k_3 = 5, k_2 = 100$ but different strategies for control points of the internal lines: by using (13) **(c)**, and by placing them on a plane **(d)**

corresponding values of the second derivatives decrease. In Fig. 2**(a)-(c)** is reported an example of a triangular element for different choices of the degrees.

We conclude with two technical lemmas useful in the next section.

Lemma 1. *Let Γ_r, $r = 1,2,3$ be the plane defined by the data (7) and let*

$$\Delta_r^+ = \{(x,y) \text{ s.t. } \Gamma_r(x,y) \geq 0\} \quad .$$

Let

$$Q_1 = P_1\,,\, Q_2 = P_1 + \frac{P_2 - P_1}{k_1}\,,\, Q_3 = P_1 + 2\frac{P_2 - P_1}{k_1} \quad ,$$

$$Q_4 = P_2 - 2\frac{P_2 - P_1}{k_2}\,,\, Q_5 = P_2 - \frac{P_2 - P_1}{k_2}, Q_6 = P_2 \quad ,$$

and let us recall the notations of (12). If

$$Q_1\,,\, Q_2\,,\, Q_3\,,\, Q_1 + \frac{P_3 - H}{\overline{k}}, Q_2 + \frac{P_3 - H}{\overline{k}}\,,\, Q_3 + \frac{P_3 - H}{\overline{k}} \in \Delta_1^+$$

and

$$Q_4\,,\,Q_5\,,\,Q_6\,,\,Q_4+\frac{P_3-H}{\overline{k}}\,,\,Q_5+\frac{P_3-H}{\overline{k}}\,,\,Q_6+\frac{P_3-H}{\overline{k}}\in\Delta_2^+\quad,$$

then the control points given by (10) and (12) are non negative.

Proof. Let us remind (10); since from the above conditions we obtain $\ell_\psi \geq 0$, the claim in the case $u_3 = 0$ follows from the fact that λ_ψ is obtained from ℓ_ψ with a corner cutting process. Similarly, by the linearity of the operators $\mathscr{E}_{\widetilde{k}}^{\overline{k}}$, $\mathscr{D}_{\mathbf{k}}^{\widetilde{k}}$, the process described in (12) can be written as

$$\mathscr{E}_{\widetilde{k}}^{\overline{k}}\mathscr{D}_{\mathbf{k}}^{\widetilde{k}}\left(\ell_\psi+\delta\right)\quad,$$

and the hypotheses of the lemma imply that $\ell_\psi + \delta$ is non negative. □

Lemma 2. *If the conditions required in Lemma 1 are fulfilled for all the edges P_1P_2, P_2P_3, P_3P_1 then all the control points are non-negative.*

Proof. Using a rotation of the vertices Lemma 1 implies that the control points (9) and (11) are non-negative. The claim follows observing that the addenda in (13) are given by linear interpolation of control points lying on the second lines. □

4 Construction of a Spanning System of Splines

Aim of this section is to study the construction of spline functions on the regular triangulation defined by the directions $(1,0)$, $(1/2,\sqrt{3}/2)$, $(1/2,-\sqrt{3}/2)$, in order to obtain a spanning system. These spline functions can be obtained by suitably collecting the triangular elements described in the above section. Their tension properties are obviously inherited by the splines.

Let $P=\{P_r;\ r=1,2,\ldots,R\}$ be a general set of planar points organized in a triangulation $\{T_\nu;\ \nu=1,2,\ldots,N\}$ and let $D=\mathrm{ConvHull}(P)=\bigcup_{\nu=1}^{N}T_\nu$. Moreover let $I_r=\{\nu \text{ s.t. } P_r\in T_\nu\}$ and $\overline{T}_{P_r}=\bigcup_{\nu\in I_r}T_\nu$, i.e. the union of triangles which share the vertex P_r. Assume we are given the Hermite conditions

$$f(P_r)\ ;\ f_x(P_r),f_y(P_r)\ ;\ f_{xx}(P_r)=f_{xy}(P_r)=f_{yy}(P_r)=0\ ;\ r=1,2,\ldots,R\quad,$$

and the degrees $k=\{k_1,k_2,\ldots,k_R\}$. We consider two adjacent triangles given by $T_{\nu_1}=\triangle(P_{r_1}P_{r_2}P_{r_3})$, $T_{\nu_2}=\triangle(P_{r_1}P_{r_4}P_{r_2})$. By construction the composite function they form is C^2 at P_{r_1} and P_{r_2}; moreover, the same formulas (10) and (12) are applied for the two triangles, ensuring both that the control points on the common boundary coincide and that the opposite points of the second lines are coplanar. Standard results on composite triangular Bézier surfaces (see, e.g. [15], p. 291) ensure us that the two triangular polynomials are C^1 across the common edge $P_{r_1}P_{r_2}$.

We remark that the degrees are independent and therefore can be chosen for satisfying local shape constraints. Note that k_r influences only the triangles in $\overline{T}_{P_r}$. Therefore, if we increase the only k_{r_1} the surface exhibit a cuspidal shape around P_{r_1} and tends locally to $f(P_{r_1})$. If we increase k_{r_1} and k_{r_2} the surface tends, along the edge P_{r_1}, P_{r_2}, to the line connecting $f(P_{r_1})$ and $f(P_{r_2})$ and, if we increase k_{r_1}, k_{r_2} and k_{r_3}, the corresponding patch tends to the plane connecting $f(P_{r_1})$, $f(P_{r_2})$ and $f(P_{r_3})$.

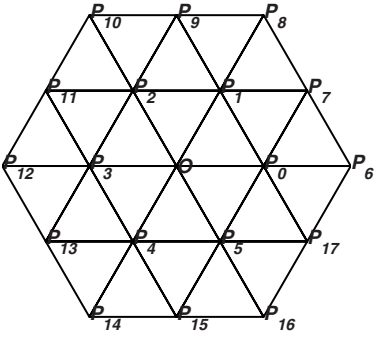


Fig. 3. The support $\widetilde{D}$

Let us now study the construction of a single spline function. Using the same notations as in the previous section, let the set of points

$$\widetilde{P} = \{O, P_0, P_1, \ldots, P_{17}\} \ , \ O = (X_O, Y_O), P_r = (X_r, Y_r) \ ,$$

be extracted from the regular triangulation as shown in Fig. 3. The set $\widetilde{D} = \text{ConvHull}(\widetilde{P})$ will constitute the support of our spline. Without loss of generality we assume $O = (0,0)$ and the edges of all triangles of $\widetilde{D}$ with unit length. In particular we have $P_{12} = (-2,0)$, $P_3 = (-1,0)$, $P_0 = (1,0)$ and $P_6 = (2,0)$. Moreover, we define the directions of the triangulations as

$$\varepsilon_i = P_i - O \ , \ i = 0, \ldots, 5 \ .$$

Let k_O, k_r, $r = 0, \ldots, 17$ be the degrees associated to $\widetilde{P}$. We use variable degree triangular elements of the form (6) and join them in order to form a positive spline which vanishes at the boundary of $\widetilde{D}$. Since the triangular elements are uniquely defined by the Hermite data (7), we specify the values of

$$f(O), f_x(O), f_y(O) \ ; \ f(P_r), f_x(P_r), f_y(P_r) \ , \ r = 0, 1, \ldots, 17 \ , \tag{14}$$

and use them to ensure that the Bézier control points satisfy the required properties. In particular, in order to define a spanning system of spline functions useful in the applications, the data (14) will be computed imposing in turn central symmetry, compact support, linear function reproduction and finally positivity and shape constraints.

If we impose that the function vanishes at the boundary of the domain $\widetilde{D}$ we obtain

$$f(P_r) = f_x(P_r) = f_y(P_r) = 0 \ , \ r = 6, \ldots, 17 \ . \tag{15}$$

The symmetry around the point O implies immediately $\langle [f_x(O), f_y(O)], \rho \rangle = 0$ for any direction ρ, that is

$$f_x(O) = f_y(O) = 0 \ ; \tag{16}$$

moreover, because of symmetry we require that

$$f(P_0) = f(P_1) = \cdots = f(P_5) \ , \tag{17}$$

and if we call $M_r,\ r=1,\dots,5$ the rotation matrices such that

$$\varepsilon_r^T = M_r \varepsilon_0^T \ ,\ r=1,\dots,5 \ ,$$

then we have to impose

$$[f_x(P_r), f_y(P_r)]^T = M_r\,[f_x(P_0), f_y(P_0)]^T \ . \tag{18}$$

In addition we must require that

$$\left\langle [f_x(P_r), f_y(P_r)], \varepsilon_1 \right\rangle = \left\langle [f_x(P_r), f_y(P_r)], \varepsilon_5 \right\rangle \ ,$$

which along with (18) gives

$$\frac{1}{2} f_x(P_0) + \frac{\sqrt{3}}{2} f_y(P_0) = \frac{1}{2} f_x(P_0) - \frac{\sqrt{3}}{2} f_y(P_0) \ ,$$

and so

$$f_y(P_0) = 0 \ . \tag{19}$$

Let us consider now the property of partition of unity and $\mathbb{P}_1$ reproduction.

From Remark 1 we know that if the Hermite data are taken from an affine function the triangular patch reproduces the affine function. Therefore, it will suffice to impose that the values f, f_x, f_y are taken from the functions $1, x$ and y. Moreover, because all our compactly supported functions can be obtained by translating the central one, we can limit ourselves to consider the sum of the Hermite data at the center O. Before, it is worth observing that the degree k_O or k_r associated with the point O or P_r is shared by all the functions not vanishing at that point. For instance, for the function centered at P_3 the point O plays the same role that the point P_0 plays for the B-spline centered at O. In other words the value of the sum of the compactly supported functions at O is given by

$$f(O) + \sum_{r=0}^{17} f(P_r) = f(O) + \sum_{r=0}^{5} f(P_r) \ ,$$

(similar formulas hold for the partial derivatives) and we assume that

$$k_O = k_0 = \cdots = k_{17} := k \ . \tag{20}$$

If we impose the partition of unity we therefore obtain

$$f(P_0) = \frac{1 - f(O)}{6} \ , \tag{21}$$

and note that (16) and (18) imply that the condition

$$[f_x(O), f_y(O)] + \sum_{r=0}^{5} [f_x(P_r), f_y(P_r)] = [0,0] \ ,$$

is automatically satisfied. We now impose that x and y are reproduced. Since we use a uniform triangulation and assume that (20) holds, we may choose $O, P_0, \ldots, P_5$ as *Greville points*. We start observing that, because of symmetry, the conditions

$$X_O f(O) + \sum_{r=0}^{5} X_r f(P_r) = X_O = 0 \ , \ Y_O f(O) + \sum_{r=0}^{5} Y_r f(P_r) = Y_O = 0 \ ,$$

are implicitly satisfied and that

$$X_O\,[f_x(O), f_y(O)] + \sum_{r=0}^{5} X_r\,[f_x(P_r), f_y(P_r)] = [1,0] \ ,$$
$$Y_O\,[f_x(O), f_y(O)] + \sum_{r=0}^{5} Y_r\,[f_x(P_r), f_y(P_r)] = [0,1] \ ,$$

lead respectively to

$$[-3f_x(P_0), 0] = [1,0] \ ; \ [0, -3f_x(P_0)] = [0,1] \ ,$$

which in turn give

$$f_x(P_0) = -\frac{1}{3} \ . \tag{22}$$

For the remaining constraints we introduce the points E, F, G. Recalling that along each direction ε_i the spline section is a piecewise curve composed by 2 quintic-like variable degree curves, in the segment OP_6, we denote with E, F the third and third but last pseudo-Bézier control points of the first piece, while G is the third control points of the second piece. We refer to Fig. 5, where we have depicted the projection of the above mentioned control points on the xy-plane.

Let us turn then to positivity constraints. We concentrate on the subset

$$\overline{D} = \mathrm{ConvHull}\left(\{P_0, P_6, P_7, P_1, O, P_5, P_{17}\}\right) \ ,$$

as shown in Fig. 4 (the other similar subsets are obtained via a central symmetry) and in particular on the control points around P_0 (see Fig. 5). We recall that $f_x(P_0) = -1/3$, $f_y(P_0) = 0$ and $f_{xx}(P_0) = 0$, therefore if the control point G is nonnegative then by construction the whole pseudo-Bézier polygon and the curve (relative to the piece P_0P_6) do the same. Recall that in our construction we are assuming (20), than $k_0 = k_6 = k$. For simplicity we can impose, Fig. 5, that

$$G = 0 \ \text{ that is } \ f(P_0) - \frac{2}{3k} = 0 \ ,$$

obtaining, from (21),

$$f(P_0) = \frac{2}{3k} \ , \ f(O) = \frac{k-4}{k} \ . \tag{23}$$

Note that the plane given by $f(P_0), f_x(P_0), f_y(P_0)$ is non-negative for $x \leq 1 + 2/k$; in particular, all the points around P_0 and included in the rectangle depicted in Fig. 5 are non-negative. We also observe that, since $f_x(O) = f_y(O) = 0$, the points around

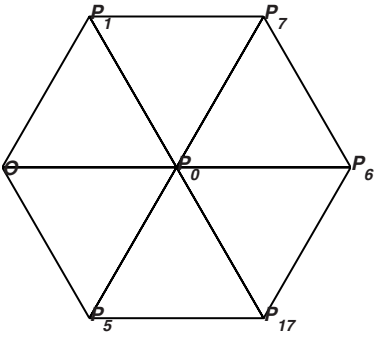


Fig. 4. The subset $\overline{D}$

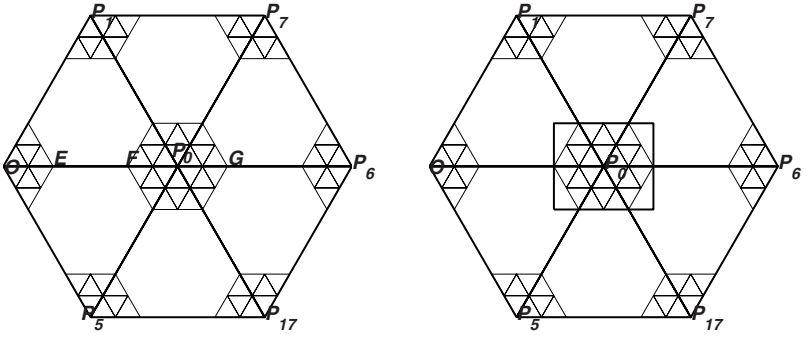


Fig. 5. Left: the projection of the control points around the vertices. Right: the rectangle containing non-negative control points.

O shown in Fig. 5 are all equal to $f(O)$. Therefore, by symmetry, the points directly obtainable from the function values, its first and its (vanishing) second derivatives are non-negative.

The non-negativity of the remaining control points follows immediately from Lemmas 1 and 2.

We refer again to Fig. 5; since we would like to have a *bell shaped* basis function, it is reasonable to impose that the value F of the control polygon is less or equal to the value E, that is $F = f(P_0) - 2f_x(P_0)/k \leq E = f(O)$, which leads to

$$\frac{2}{3k} + \frac{1}{3}\frac{2}{k} \leq \frac{k-4}{k} \quad ;$$

the above inequality is satisfied for

$$k \geq 6 \quad . \tag{24}$$

Therefore we shall assume that all the degrees are greater or equal to six, even though this fulfills only *visually pleasing* requirements.

We will denote with $\widetilde{\varphi}$ the C^1 variable degree polynomial spline defined by (16)–(24). We introduce the bi-infinite regular triangulation $\mathscr{T}$ defined by the vertices

$$P = \left\{ P_\gamma = \left(X_\gamma, Y_\gamma \right) \ ; \ \gamma = (\gamma_1, \gamma_2) \in \mathbb{Z}^2 \right\} \quad , \tag{25}$$

where

$$X_\gamma = \begin{cases} \gamma_1 & \text{for } \gamma_2 \text{ even} \\ \gamma_1 - \frac{1}{2} & \text{for } \gamma_2 \text{ odd} \end{cases} \ ; \ Y_\gamma = \frac{\sqrt{3}}{2} \gamma_2 \quad ,$$

and the associated sequence of degrees

$$k = \{ k_\gamma \, ; \, k_\gamma \geq 6, \ \gamma \in \mathbb{Z}^2 \ \}.$$

Referring to Fig. 1, we note that $O = P_{(0,0)}$ and we denote with D_γ the translated of $\widetilde{D}$ and, setting $\varphi_{(0,0)} = \widetilde{\varphi}$, with φ_γ the C^1 spline function defined by formulas (16)–(24) applied to D_γ.

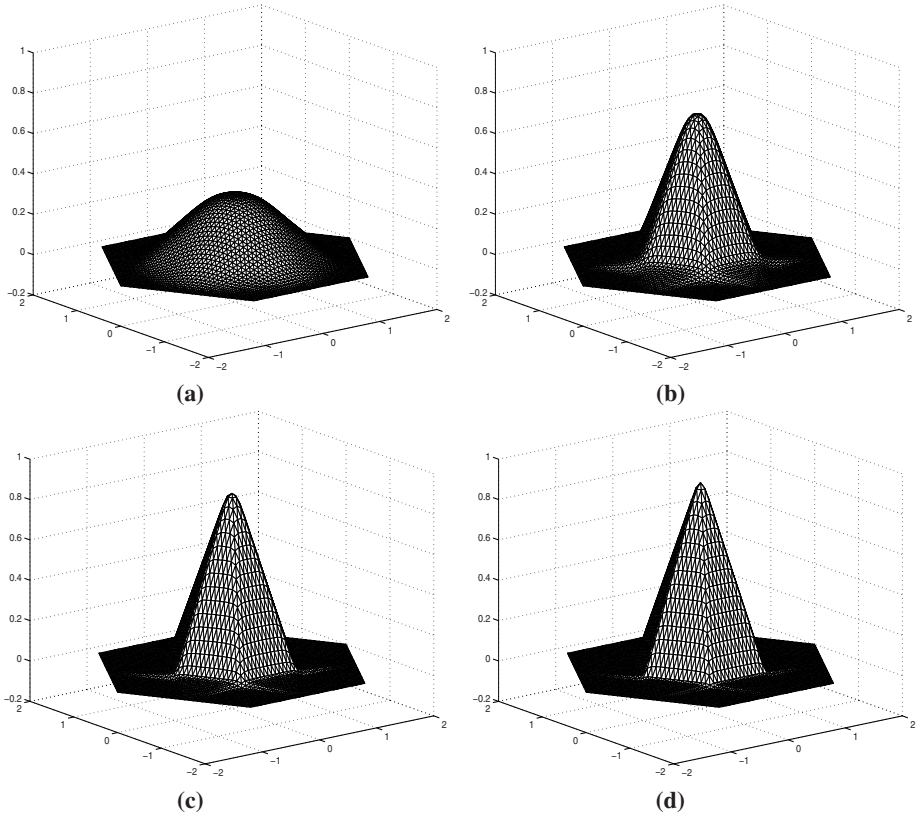


Fig. 6. Some basis functions. **(a)** $k_O = k_i = 6$ all i; **(b)** $k_O = k_i = 15$ all i; **(c)** $k_O = k_i = 30$ all i; **(d)** $k_O = k_i = 50$ all i.

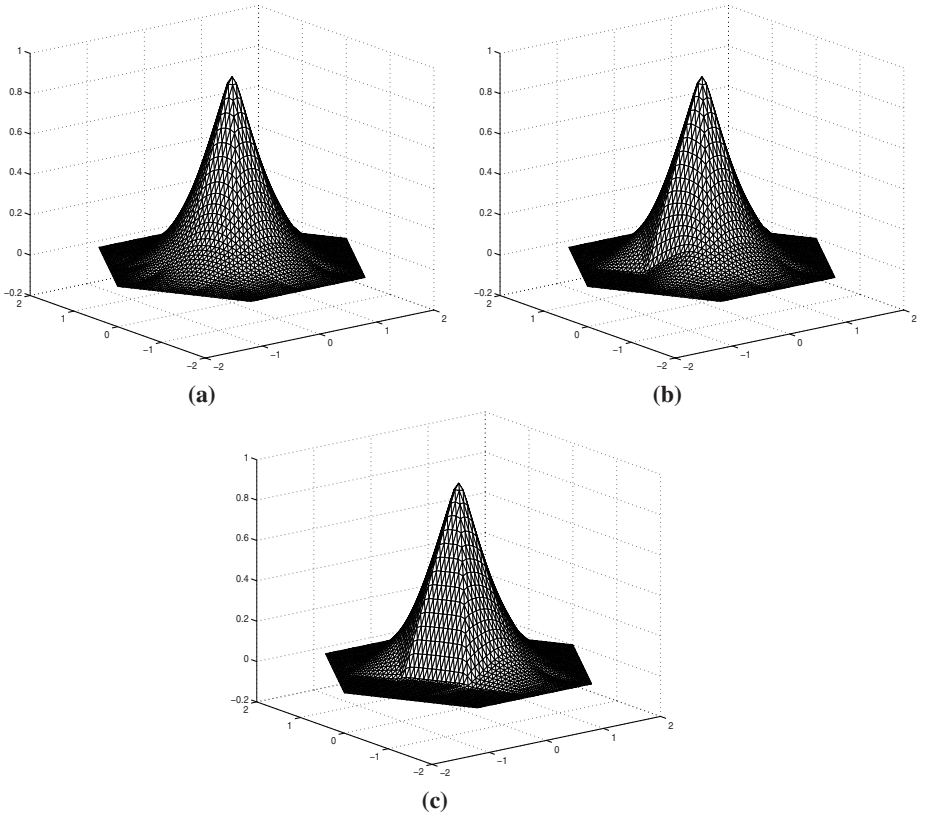


Fig. 7. Some basis functions. $k_Q = k_i = 6$, all i, except, **(a)**: $k_O = 50$; **(b)** $k_O = k_3 = 50$; **(c)** $k_O = k_3 = k_4 = 50$.

We have the following result.

Theorem 4. *The functions φ_γ have the following properties*

- $\varphi_\gamma \in C^1\left(\mathbb{R}^2\right)$ *and* $\varphi_\gamma \in C^2\left(P_\gamma\right)$;
- $\varphi_\gamma \geq 0$ *and* $\varphi_\gamma(x,y) > 0$ *for* $(x,y) \in \mathrm{Int}\left(D_\gamma\right)$;
- $\sum_{\gamma\in\mathbb{Z}^2} \varphi_\gamma = 1$; $\sum_{\gamma\in\mathbb{Z}^2} X_\gamma \varphi_\gamma = x$; $\sum_{\gamma\in\mathbb{Z}^2} Y_\gamma \varphi_\gamma = y$.

We refer to Figs. 6 and 7 for graphical examples of the functions φ_γ for different degrees. We conclude this section with some remarks.

Remark 2. *For any triangle T of the uniform triangulation (25) there are 12 non-vanishing B-splines φ_γ on T. Since the finite element space defined in Subsection 2.2 has dimension 9, it follows that the functions φ_γ are not linearly independent if restricted to T. For similar results in the standard polynomial case see [1, 4, 17, 20].*

Remark 3. *In principle, it is possible to construct compactly supported spline functions on a six-triangle support (referring to Fig. 7, on* $\mathrm{ConvHull}(O,P_0,\ldots,P_5)$*). However in this case we would have vanishing gradients at any points* P_γ*, with unpleasant shapes and without the* $\mathbb{P}_1$ *reproduction.*

Remark 4. *In view of their applications in quasi-interpolation, one of the typical requests for compactly supported splines is the* $\mathbb{P}_2$ *reproduction. In the present case we are interested in tension properties and, while the reproduction of* $\mathbb{P}_1$ *is a necessary request, the* $\mathbb{P}_2$ *reproduction would be in contrast with our goals.*

5 Examples and Conclusions

The aim of this section is to present the simplest application of the compactly supported functions, namely in quasi-interpolation. Let the functions φ_γ, $\gamma \in \mathbb{Z}^2$ be defined as in the previous section, and let us introduce the space

$$VS_k = \mathrm{span}\{\varphi_\gamma : \gamma \in \mathbb{Z}^2\} \ .$$

and, for any function $f \in C(\mathbb{R}^2)$, the quasi-interpolation operator

$$Q : C(\mathbb{R}^2) \mapsto VS_k \ , \ \ Qf = \sum_\gamma f(P_\gamma)\,\varphi_\gamma \ . \tag{26}$$

Let $T \in \mathscr{T}$ be any triangle of the regular triangulation, and let $(x,y) \in T$; since the splines φ_γ have the compact support D_γ, it is immediate to see that the evaluation of $Qf(x,y)$ requires the evaluation of the twelve basis splines centered at the points specified in Fig. 8. Let F be the continuous piecewise linear function such that

$$F(P_\gamma) = f(P_\gamma) \ ;$$

from the asymptotic properties of the spline functions φ_γ we immediately have the following results.

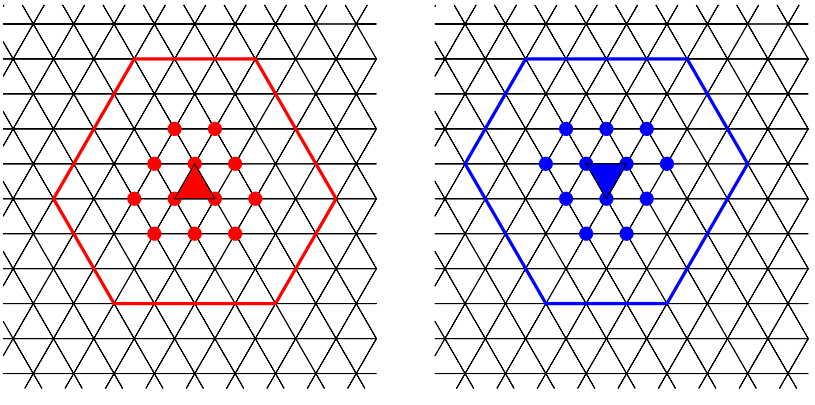


Fig. 8. The centers of supports of basis functions φ_γ needed for the computation of $Qf(x,y)$

Proposition 1. *Let the quasi-interpolation operator Qf be given by (26), and let F be the piecewise linear function interpolating the data. Then*

$$\text{if } k_\gamma \to \infty \text{ then } |Qf(P_\gamma) - f(P_\gamma)| \to 0 \ ;$$

$$\text{if } k_\gamma \to \infty \ \forall \gamma \text{ then } \|Qf - F\|_\infty \to 0 \ .$$

In practical applications, the quasi-interpolant Qf can be used to model the object described by the function F. Figure 9 shows an example in the parametric setting. We end this section with some conclusive remarks and open questions.

We have presented the construction of a new family of compactly supported splines with tension properties on a regular triangulation. These functions have been obtained by joining triangular elements with tension properties and the construction of the triangular elements is done by using univariate results and without triangular splits. Finally, applications to quasi-interpolation have been shown.

The first open question concerns a *degree-dependent* definition of the constraints (17)–(19) which takes into account the subsequent modifications on the shape of the compactly supported function. The second point is given by second derivatives. In principle, it is possible to use non-vanishing Hessian matrices at the knots. However, while the changes induced in the symmetry constraints seem to be manageable, the positivity constraints appear much more involved. The last question is a classical one and concerns a possible generalization to general triangulations –which would greatly enlarge the practical applications of our method. Indeed, this problem is very hard also in the case of polynomial box splines, and we are not optimistic about its solution.

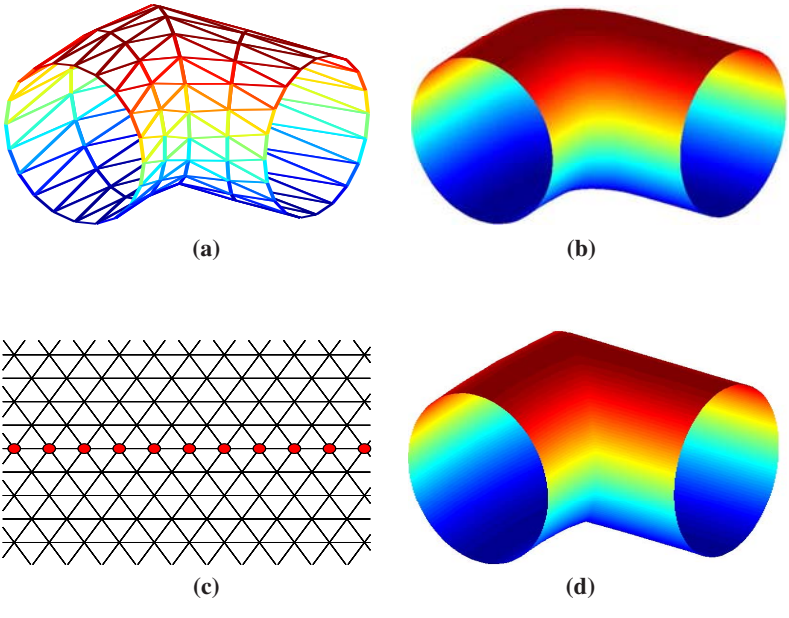


Fig. 9. An example of parametric quasi-interpolant. **(a)** the data surface. **(b)** Quasi-interpolant, $k_\gamma = 6$ all γ. **(c)** Some degrees have been increased to 40. **(d)** The resulting quasi-interplant.

Acknowledgment. We would like to thank the referees for their helpful comments and remarks.

References

1. De Boor, C., Höllig, K.: Bivariate box splines and smooth pp functions on a three direction mesh. J. Comput. Applied Math. 9, 13–23 (1983)
2. De Boor, C.: Multivariate piecewise polynomials. Acta Numerica, 65–109 (1993)
3. Carnicer, J.M., Peña, J.M.: Tchebycheff spaces and total positivity. In: Gasca, M., Micchelli, C.A. (eds.) Total positivity and its applications, pp. 277–300. Kluwer, Dordrecht (1996)
4. Chui, C.K.: Multivariate Splines. CBMS-NSF Regional Conference Series in Appl. Math, vol. 54. SIAM, Philadelphia (1988)
5. Chui, C.K., Lai, M.L.: Computation of box splines and B-splines on triangulations of nonuniform rectangular partitions. Approx. Theory Appl. 3, 37–62 (1987)
6. Chui, C.K., Lai, M.L.: Algorithms for generating B-nets and graphically displaying spline surfaces on three- and four-directional meshes. Comput. Aided Geom. Design 8, 479–493 (1991)
7. Costantini, P.: Curve and surface construction using variable degree polynomial splines. Comput. Aided Geom. Design 24, 426–446 (2000)
8. Costantini, P., Manni, C.: A local tension scheme for scattered data interpolation. Comput. Aided Geom. Design 16, 385–405 (1999)
9. Costantini, P., Manni, C.: A parametric cubic element with tension properties. SIAM J. Numer. Anal. 36, 607–628 (1999)
10. Costantini, P., Manni, C.: Interpolating polynomial macro-elements with tension properties. In: Curve and Surface Fitting, Saint-Malo 1999, pp. 143–152. Nashboro Press, Brentwood (2000)
11. Costantini, P., Manni, C.: Shape-preserving C^3 interpolation: the curve case. Adv. Comp. Math. 18, 41–63 (2003)
12. Costantini, P., Pelosi, F.: Shape preserving approximation of spatial data. Adv. Comp. Math. 20, 25–51 (2004)
13. Costantini, P., Pelosi, F., Sampoli, M.L.: Triangular surface patches with shape constraints. In: Chenin, P., Lyche, T., Schumaker, L.L. (eds.) Curve and Surface Design: Avignon 2006, pp. 123–132. Nashboro Press, Brentwood (2007)
14. Dierckx, P.: On Calculating normalized Powell-Sabin B-splines. Comput. Aided Geom. Design 15, 61–78 (1997)
15. Farin, G.: Curves and Surfaces for Computer Aided Geometric Design, 4th edn. Academic Press, London (1996)
16. Goodman, T.N.T.: Total positivity and the shape of curves. In: Gasca, M., Micchelli, C.A. (eds.) Total Positivity and its Applications, pp. 157–186. Kluwer, Dordrecht (1996)
17. Lai, M.J., Schumaker, L.L.: Splines on triangulations. Cambridge University Press, Cambridge (2007)
18. Mazure, M.-L.: Chebyshev spaces and Bernstein bases. Constr. Approx. 22, 347–363 (2005)
19. Nouisser, O., Sbibih, D., Sablonnière, P.: Pairs of B-splines with small support on the four-directional mesh generating a partition of unity. Adv. Comput. Math. 21, 317–355 (2004)
20. Sablonnière, P.: A Catalog of B-splines of degree ≤ 10 on a three direction mesh, Universitè de Lille (1984)
21. Schumaker, L.L.: Spline Functions: Basic Theory. Wiley, New York (1981)
22. Schumaker, L.L.: On super splines and finite elements. SIAM J. Numer. Anal. 36, 997–1005 (1989)

Some Geometrical Aspects of Control Points for Toric Patches

Gheorghe Craciun[1], Luis David García-Puente[2], and Frank Sottile[3,⋆]

[1] Department of Mathematics and Department of Biomolecular Chemistry, University of Wisconsin, Madison, WI 53706, USA
craciun@math.wisc.edu
www.math.wisc.edu/~craciun

[2] Department of Mathematics and Statistics, Sam Houston State University, Huntsville, TX 77341, USA
lgarcia@shsu.edu
www.shsu.edu/~ldg005

[3] Department of Mathematics, Texas A&M University, College Station, TX 77843, USA
sottile@math.tamu.edu
www.math.tamu.edu/~sottile

Abstract. We use ideas from algebraic geometry and dynamical systems to explain some ways that control points influence the shape of a Bézier curve or patch. In particular, we establish a generalization of Birch's Theorem and use it to deduce sufficient conditions on the control points for a patch to be injective. We also explain a way that the control points influence the shape via degenerations to regular control polytopes. The natural objects of this investigation are irrational patches, which are a generalization of Krasauskas's toric patches, and include Bézier and tensor product patches as important special cases.

Introduction

The control points and weights of a Bézier curve, Bézier patch, or tensor-product patch govern many aspects of the curve or surface. For example, they provide an intuitive means to control its shape. Through de Castlejau's algorithm, they enable the computation of the curve or surface patch. Finer aspects of the patch, particularly continuity and smoothness at the boundary of two patches are determined by the control points and weights. Global properties, such as the location of a patch in space due to the convex hull property, also depend upon the control points. When the control points are in a particular convex position, then the patch is convex [1].

We apply methods from algebraic geometry, specifically toric geometry, to explain how some further global properties of a patch are governed by the control points. We first investigate the self-intersection, or injectivity of a patch. We give a simple and easy-to-verify condition on a set of control points which implies that the resulting patch has no self-intersection, for any choice of weights. For 3-dimensional patches as used

⋆ This material is based in part upon work supported by the Texas Advanced Research Program under Grant No. 010366-0054-2007 and NSF grant DMS-070105.

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 111–135, 2010.

for solid modeling, injectivity is equivalent to the patch properly parameterizing the given solid. This uses Craciun and Feinberg's injectivity theorem [2] from the theory of chemical reaction networks, which may be seen as a generalization of Birch's Theorem from algebraic statistics.

A second global property that we investigate is how the shape of the patch is related to the shape of a *control polytope*. This is a piecewise linear triangulated surface whose vertices are the control points. It is *regular* if the underlying triangulation comes from a regular triangulation of the domain polytope of the patch. We show that regular control polytopes are the limits of patches as the weights undergo a toric deformation corresponding to the underlying regular triangulation, and that non-regular control polytopes can never be such a limit. This gives a precise meaning to the notion that the shape of the control net governs the shape of the patch.

This line of inquiry is pursued in terms of Krasauskas's toric patches [3], as it relies upon the structure of toric varieties from algebraic geometry. The correct level of generality is however that of *irrational (toric) patches*, which are analytic subvarieties of the simplex (realized as a compactified positive orthant) that are parameterized by monomials x^α, where x is a vector of positive numbers and the exponent vector α has real-number coordinates. (This is a usual toric variety when α has integer coordinates.) While irrational patches may seem exotic for modeling, they occur naturally in statistics as discrete exponential families [4] and their blending functions may be computed using iterative proportional fitting (IPF) [5], a popular numerical algorithm from statistics. Furthermore, these blending functions have linear precision. For toric patches, this was observed in [6] and developed in [7], and the analysis there carries over to irrational patches. While we work in this generality, our primary intent (and the main application) is to shed light on properties of Bézier curves, surfaces, and 3-dimensional patches.

We recall the standard definition of a mapping via control points and blending functions, and then the definitions of toric Bézier patches in Sect. 1. There, we also illustrate some of our results on examples of Bézier curves. In Sect. 2, we introduce irrational toric patches, recalling the geometric formulation of a toric patch and the use of iterative proportional fitting to compute these patches, explaining how these notions from [7] for toric patches extend to irrational patches. The next two sections contain our main results. We study injectivity of patches in Sect. 3, and discuss degenerations to control polytopes in Sect. 4. Appendices A and B contain technical proofs of some theorems.

1 Toric Bézier Patches

We interpret the standard definition of a mapping via control points and blending functions (see for example [8, §2]) in a general form convenient for our discussion. All functions here are smooth (C^∞) where defined and real-valued. Let $\mathbb{R}_>$ be the set of strictly positive real numbers and $\mathbb{R}_\geq$ the set of non-negative real numbers. We will use the following typographic conventions throughout. Vector constants (control points, indexing exponents, and standard basis vectors) will be typeset in **boldface**, while vector variables will be typeset in standard math italics.

Let $\mathscr{A}$ be a finite set of points that affinely span $\mathbb{R}^d$, which we shall use as geometrically meaningful indices. A control point scheme for parametric patches, or (*parametric*) *patch*, is a collection $\beta = \{\beta_{\mathbf{a}} \mid \mathbf{a} \in \mathscr{A}\}$ of non-negative functions, called *blending*

functions. The common domain of the blending functions is the convex hull Δ of $\mathscr{A}$, which we call the *domain polytope*. We also assume that the blending functions do not vanish simultaneously at any point of Δ, so that there are no basepoints.

Lists $\mathscr{B} := \{\mathbf{b}_\mathbf{a} \mid \mathbf{a} \in \mathscr{A}\} \subset \mathbb{R}^n$ of *control points* and positive *weights* $w := \{w_\mathbf{a} \in \mathbb{R}_> \mid \mathbf{a} \in \mathscr{A}\} \in \mathbb{R}_>^{\mathscr{A}}$ together give a map $F : \Delta \to \mathbb{R}^n$ defined by

$$F(x) \;:=\; \frac{\sum_{\mathbf{a}\in\mathscr{A}} w_\mathbf{a}\beta_\mathbf{a}(x)\,\mathbf{b}_\mathbf{a}}{\sum_{\mathbf{a}\in\mathscr{A}} w_\mathbf{a}\beta_\mathbf{a}(x)} \quad . \tag{1}$$

The denominator in (1) is positive on Δ and so the map F is well-defined.

Remark 1. We will refer to both $\beta_\mathbf{a}(x)$ and $w_\mathbf{a}\beta_\mathbf{a}(x)$ as the blending functions of a patch. This generality of separating the weights from the blending functions will be used in Sect. 4 when we investigate the effect of systematically varying the weights of a patch while keeping the control points and blending functions constant.

The control points and weights affect the shape of the patch which is the image of the map F (1). For example, the *convex hull property* asserts that the image $F(\Delta)$ of the patch lies in the convex hull of the control points. To see this, note that if we set

$$\overline{\beta}_\mathbf{a}(x) \;:=\; \frac{w_\mathbf{a}\beta_\mathbf{a}(x)}{\sum_{\mathbf{a}\in\mathscr{A}} w_\mathbf{a}\beta_\mathbf{a}(x)} \quad ,$$

then $\overline{\beta}_\mathbf{a}(x) \geq 0$ and $1 = \sum_{\mathbf{a}\in\mathscr{A}} \overline{\beta}_\mathbf{a}(x)$. Then formula (1) becomes

$$F(x) \;=\; \sum_{\mathbf{a}\in\mathscr{A}} \overline{\beta}_\mathbf{a}(x)\,\mathbf{b}_\mathbf{a} \quad ,$$

so that $F(x)$ is a convex combination of the control points and therefore lies in their convex hull. In fact, if there is a point $x \in \Delta$ at which no blending function vanishes, then *any* point in the interior of the convex hull of the control points is the image $F(x)$ of some patch for some choice of weights. In this way, the convex hull property is the strongest general statement that can be made about the location of a patch.

Another well-known manifestation of control points is the relation of a Bézier curve to its control polygon. Fix a positive integer m and let $\mathscr{A} := \{\frac{i}{m} \mid i = 0,\ldots,m\}$ so that Δ is the unit interval. The blending functions of a Bézier curve are the Bernstein polynomials,

$$\beta_i(x)\;(=\;\beta_{\frac{i}{m}}(x)) \;:=\; \tbinom{m}{i}x^i(1-x)^{m-i} \quad .$$

The *control polygon* of a Bézier curve with control points $\mathbf{b}_0,\mathbf{b}_1,\ldots,\mathbf{b}_m$ is the union of the line segments $\overline{\mathbf{b}_0,\mathbf{b}_1}, \overline{\mathbf{b}_1,\mathbf{b}_2}, \ldots, \overline{\mathbf{b}_{m-1},\mathbf{b}_m}$ between consecutive control points. Fig. 1 displays two quintic plane Bézier curves with their control polygons (solid lines). The convex hulls of the control points are indicated by the dashed lines. The first curve has no points of self-intersection, while the second curve has one point of self-intersection. While this self-intersection may be removed by varying the weights attached to the control points, by Theorem 11 it is impossible to find weights so that a curve with the first set of control points has a point of self-intersection.

We will also show that the control polygon may be approximated by a Bézier curve. We state a simplified version of Theorem 13 from Sect. 4.

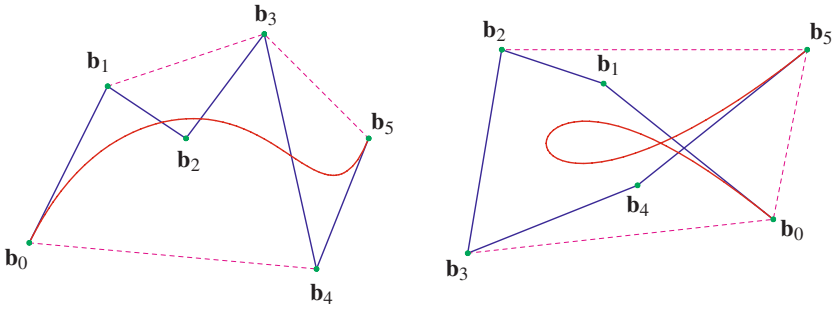


Fig. 1. Quintic Bézier curves

Theorem. *Given control points in $\mathbb{R}^n$ for a Bézier curve and some number $\varepsilon > 0$, there is a choice of weights so that the image $F[0,1]$ of the Bézier curve lies within a distance ε of the control polygon.*

In Fig. 2, we display one of the quintic curves from Fig. 1, but with weights on $\mathbf{b}_0$—$\mathbf{b}_5$ of $(1, 20^2, 20^3, 20^3, 20^2, 1)$ and $(1, 300^2, 300^3, 300^3, 300^2, 1)$, respectively. The control polygon for the second curve is omitted, as it would obscure the curve. The first curve lies within a distance $\varepsilon = 0.13$ of the control polygon and the second within a distance $\varepsilon = 0.02$, if the control polygon has height 1.

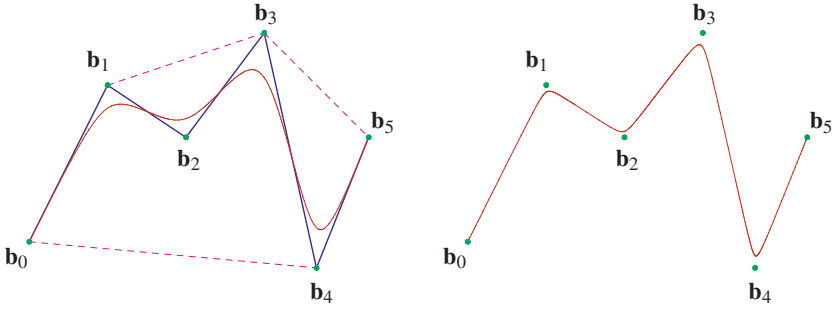


Fig. 2. Degenerating quintics

1.1 Toric Patches

Krasauskas [3] introduced toric patches as a generalization of the classical Bézier and tensor product patches. These are based upon toric varieties from algebraic geometry and their shape may be any polytope with integer vertices. The articles [9,6] provide an introduction to toric varieties for geometric modeling.

A polytope Δ is defined by its *facet inequalities*

$$\Delta \;=\; \{x \in \mathbb{R}^d \mid 0 \leq h_i(x), i = 1, \ldots, \ell\} \;.$$

Here, Δ has ℓ facets (faces of maximal dimension) and for each $i = 1, \ldots, \ell$, $h_i(x) = \mathbf{v}_i \cdot x + c_i$ is the linear function defining the ith facet, where $\mathbf{v}_i \in \mathbb{Z}^d$ is the (inward oriented) primitive vector normal to the facet and $c_i \in \mathbb{Z}$.

For example, if our polytope is the triangle with vertices $(0,0)$, $(m,0)$, and $(0,m)$,

$$m\triangle \;:=\; \{(x,y) \;\in\; \mathbb{R}^2 \mid 0 \le x,y, \quad \text{and} \quad 0 \le m-(x+y)\} \;, \tag{2}$$

then we have $h_1 = x$, $h_2 = y$, and $h_3 = m - x - y$. Here, $m\triangle$ is the unit triangle $\triangle$ with vertices $(0,0)$, $(1,0)$, and $(0,1)$ scaled by a factor of m.

Let $\mathscr{A} \subset \Delta \cap \mathbb{Z}^d$ be any subset of the integer points of Δ which includes its vertices. For every $\mathbf{a} \in \mathscr{A}$, Krasauskas defined the *toric Bézier function*

$$\beta_{\mathbf{a}}(x) \;:=\; h_1(x)^{h_1(\mathbf{a})} h_2(x)^{h_2(\mathbf{a})} \cdots h_\ell(x)^{h_\ell(\mathbf{a})} \;, \tag{3}$$

which is non-negative on Δ, and the collection of all $\beta_{\mathbf{a}}$ has no common zeroes on Δ. These are blending functions for the *toric patch* of shape $\mathscr{A}$. If we choose weights $w \in \mathbb{R}^{\mathscr{A}}$ and multiply the formula (3) by $w_{\mathbf{a}}$, we obtain blending functions for the toric patch of shape $(\mathscr{A}, w)$.

Example 2 (Bézier triangles). When Δ is a scaled triangle or a product of such triangles, (3) gives the blending functions of the Bézier patch or Bézier simploid [10] with the corresponding shape. To see this for the scaled triangle $m\triangle$ (2), note that given an integer point $\mathbf{a} = (i,j) \in m\triangle$, and weight the multinomial coefficient $w_{(i,j)} := \frac{m!}{i!j!(m-i-j)!}$, then the corresponding blending function is

$$\beta_{(i,j)}(x,y) \;=\; \tfrac{m!}{i!j!(m-i-j)!} x^i y^j (m-x-y)^{m-i-j} \;.$$

This is almost the bivariate Bernstein polynomial, which is obtained by substituting mx and my for x and y, respectively, and dividing by m^m. (This has the effect of changing the domain from $m\triangle$ to the unit triangle $\triangle$.)

2 Irrational Patches

Krasauskas's definition (3) of toric Bézier functions still makes sense if we relax the requirement that the points $\mathscr{A} \subset \mathbb{R}^d$ have integer coordinates. This leads to the notion of an irrational patch (as its blending functions are no longer rational functions), which provides the level of generality appropriate for our investigation.

Let $\mathscr{A} \subset \mathbb{R}^d$ be a finite collection of points and set $\Delta \subset \mathbb{R}^d$ to be the convex hull of $\mathscr{A}$, which we assume is a full-dimensional polytope. We may also realize Δ as an intersection of half-spaces through its facet inequalities,

$$\Delta \;=\; \{x \in \mathbb{R}^d \mid h_i(x) \ge 0 \text{ for each } i = 1, \ldots, \ell\} \;. \tag{4}$$

Here, Δ has ℓ facets with the ith facet supported by the affine hyperplane $h_i(x) = 0$ where $h_i(x) = \mathbf{v}_i \cdot x + c_i$ with $c_i \in \mathbb{R}$ and $\mathbf{v}_i$ *an* inward pointing normal vector to the ith facet of Δ. There is no canonical choice for these data; multiplying a pair $(\mathbf{v}_i, c_i)$ by a positive scalar gives another pair defining the same half-space.

Following Krasauskas, we provisionally define *(irrational) toric Bézier functions* $\{\beta_{\mathbf{a}} \colon \Delta \to \mathbb{R}_{\geq} \mid \mathbf{a} \in \mathscr{A}\}$ by the same formula as (3),

$$\beta_{\mathbf{a}}(x) \;:=\; h_1(x)^{h_1(\mathbf{a})} h_2(x)^{h_2(\mathbf{a})} \cdots h_\ell(x)^{h_\ell(\mathbf{a})} \;.$$

These are blending functions for the *irrational toric patch* of shape $\mathscr{A}$. While these functions do depend upon the choice of data $(\mathbf{v}_i, c_i)$ for the facet inequalities defining Δ, we will see that the image $F(\Delta)$ of such a patch given by weights and control points is independent of these choices.

2.1 Geometric Formulation of a Patch

We follow Sect. 2.2 of [7], but drop the requirement that our objects are algebraic. Let $\mathscr{A} \subset \mathbb{R}^d$ be a finite subset indexing a collection of blending functions $\{\beta_{\mathbf{a}} \colon \Delta \to \mathbb{R}_{\geq} \mid \mathbf{a} \in \mathscr{A}\}$, where Δ is the convex hull of $\mathscr{A}$. Let $\mathbb{R}^{\mathscr{A}}$ be a real vector space with basis $\{e_{\mathbf{a}} \mid \mathbf{a} \in \mathscr{A}\}$. Set $\mathbb{R}^{\mathscr{A}}_{\geq} \subset \mathbb{R}^{\mathscr{A}}$ to be the points with non-negative coordinates and let $\mathbb{R}^{\mathscr{A}}_{>}$ be those points with strictly positive coordinates.

For $z = (z_{\mathbf{a}} \mid \mathbf{a} \in \mathscr{A}) \in \mathbb{R}^{\mathscr{A}}_{\geq}$, set $\sum z := \sum_{\mathbf{a} \in \mathscr{A}} z_{\mathbf{a}}$. The *$\mathscr{A}$-simplex*, $\triangle^{\mathscr{A}} \subset \mathbb{R}^{\mathscr{A}}_{\geq}$, is the set

$$\triangle^{\mathscr{A}} \;:=\; \{z \in \mathbb{R}^{\mathscr{A}}_{\geq} : \textstyle\sum z = 1\} \;.$$

We introduce homogeneous coordinates for $\triangle^{\mathscr{A}}$. If $z \in \mathbb{R}^{\mathscr{A}}_{\geq} \setminus \{0\}$, then we set

$$[z_{\mathbf{a}} \mid \mathbf{a} \in \mathscr{A}] \;:=\; \frac{1}{\sum z}(z_{\mathbf{a}} \mid \mathbf{a} \in \mathscr{A}) \;\in\; \triangle^{\mathscr{A}} \;.$$

The blending functions $\{\beta_{\mathbf{a}} \colon \Delta \to \mathbb{R}_{\geq} \mid \mathbf{a} \in \mathscr{A}\}$ give a C^∞ map,

$$\begin{aligned} \beta \;:\; \Delta &\longrightarrow \triangle^{\mathscr{A}} \\ x &\longmapsto [\beta_{\mathbf{a}}(x) \mid \mathbf{a} \in \mathscr{A}] \,. \end{aligned}$$

The reason for this definition is that a mapping $F \colon \Delta \to \mathbb{R}^n$ (1) given by the blending functions β, weights w, and control points $\mathscr{B}$ factors through the map $\beta \colon \Delta \to \triangle^{\mathscr{A}}$. To see this, first note that the weights $w \in \mathbb{R}^{\mathscr{A}}_{>}$ act on $\triangle^{\mathscr{A}}$: If $z = [z_{\mathbf{a}} \mid \mathbf{a} \in \mathscr{A}] \in \triangle^{\mathscr{A}}$, then

$$w.z \;:=\; [w_{\mathbf{a}} z_{\mathbf{a}} \mid \mathbf{a} \in \mathscr{A}] \;. \tag{5}$$

The control points $\mathscr{B}$ define the map $\pi_{\mathscr{B}} \colon \triangle^{\mathscr{A}} \to \mathbb{R}^n$ via

$$\pi_{\mathscr{B}} \;:\; z \;=\; (z_{\mathbf{a}} \mid \mathbf{a} \in \mathscr{A}) \;\longmapsto\; \sum_{\mathbf{a} \in \mathscr{A}} z_{\mathbf{a}} \mathbf{b}_{\mathbf{a}} \;.$$

Then the mapping F (1) is simply the composition

$$\Delta \xrightarrow{\;\beta\;} \triangle^{\mathscr{A}} \xrightarrow{\;w.\;} \triangle^{\mathscr{A}} \xrightarrow{\;\pi_{\mathscr{B}}\;} \mathbb{R}^n \;. \tag{6}$$

In this way, we see that the image $\beta(\Delta) \subset \triangle^{\mathscr{A}}$ of Δ under the map β determines the shape of the patch $F(\Delta)$ (1). Internal structures of the patch, such as the mapping of

texture, are determined by how β maps Δ to $\beta(\Delta)$. For example, precomposing β with any homeomorphism of Δ gives blending functions with the same image in $\triangle^{\mathscr{A}}$, but with a different internal structure.

For an irrational toric patch of shape $\mathscr{A}$ with blending functions (3), the image $\beta(\Delta) \subset \triangle^{\mathscr{A}}$ is independent of the choice of normal vectors. For this, we first define the map $\varphi_{\mathscr{A}} \colon \mathbb{R}^d_> \to \triangle^{\mathscr{A}}$ by

$$\varphi_{\mathscr{A}} \;:\; (x_1, \ldots, x_d) \longmapsto [x^{\mathbf{a}} \;:\; \mathbf{a} \in \mathscr{A}] \;. \tag{7}$$

Let $X_{\mathscr{A}}$ be the closure of the image of the map $\varphi_{\mathscr{A}}$. When $\mathscr{A} \subset \mathbb{Z}^d$, this is the positive part [11, §4] of the toric variety parameterized by the monomials of $\mathscr{A}$. When $\mathscr{A}$ is not integral, we call $X_{\mathscr{A}}$ the *(irrational) toric variety* parameterized by monomials in $\mathscr{A}$.

In Appendix B we prove the following theorem.

Theorem 3. *Suppose that $\mathscr{A} \subset \mathbb{R}^d$ is a finite set of points with convex hull Δ. Let $\beta = \{\beta_{\mathbf{a}} \mid \mathbf{a} \in \mathscr{A}\}$ be a collection of irrational toric Bézier functions for $\mathscr{A}$. Then $\beta(\Delta) = X_{\mathscr{A}}$, the closure of the image of $\varphi_{\mathscr{A}}$.*

We prove this by showing that the restriction of the map β to the interior Δ° of Δ factors through the map $\varphi_{\mathscr{A}}$.

By Theorem 3, the image of the irrational toric blending functions for $\mathscr{A}$ depends upon $\mathscr{A}$ and not upon the choice of toric blending functions for $\mathscr{A}$. Thus the shape of the corresponding patch $F(\Delta)$ (1) depends only upon $\mathscr{A}$, the weights w, and the control points $\mathscr{B}$. However, the actual parameterization of $X_{\mathscr{A}}$ by Δ, and hence of $F(\Delta)$ does depend upon the choice of toric blending functions for $\mathscr{A}$.

To ensure that the patch $F(\Delta)$ has shape reflecting that of Δ, we require that the map $\beta \colon \Delta \to X_{\mathscr{A}}$ be injective. This also guarantees that the patch $F(\Delta)$ is typically an immersion. In the context of irrational toric patches, this injectivity is guaranteed by Birch's Theorem from algebraic statistics. For a standard reference, see [12, p. 168]. When $\mathscr{A} \subset \mathbb{Z}^d$, Birch's Theorem follows from general results on the moment map in symplectic geometry [11, §4.2 and Notes to Chapter 4, p. 140].

Theorem 4 (Birch's Theorem). *Suppose $\mathscr{A} \subset \mathbb{R}^d$ is finite and let β be a collection of toric Bézier functions. If we choose control points $\mathscr{B}$ to be the corresponding points of $\mathscr{A}$, $\{\mathbf{b}_{\mathbf{a}} = \mathbf{a} \mid \mathbf{a} \in \mathscr{A}\}$, then the composition*

$$\Delta \xrightarrow{\beta} X_{\mathscr{A}} \xrightarrow{\pi_{\mathscr{B}}} \mathbb{R}^d$$

is a homeomorphism onto Δ.

By Birch's Theorem and Theorem 3, any two sets β, β' of toric Bézier functions of shape $\mathscr{A}$ differ only by a homeomorphism $h \colon \Delta \xrightarrow{\sim} \Delta$ of the polytope Δ, so that $\beta' = \beta \circ h$. In fact, h restricts to a homeomorphism on all faces of Δ. As we are concerned with the shape of a patch and not its internal structure, we follow Krasauskas' lead and make the following definition.

Definition 5. *A(n irrational) toric patch of shape $\mathscr{A}$ is any set of blending functions $\beta := \{\beta_{\mathbf{a}} \colon \Delta \to \mathbb{R}_{\geq} \mid \mathbf{a} \in \mathscr{A}\}$ such that the map $\beta \colon \Delta \to X_{\mathscr{A}}$ is a homeomorphism.*

The projection map $\pi_{\mathscr{B}}\colon \triangle^{\mathscr{A}} \to \mathbb{R}^d$ appearing in Birch's Theorem induced by the choice $\mathscr{B} = \{\mathbf{b}_\mathbf{a} = \mathbf{a} \mid \mathbf{a} \in \mathscr{A}\}$ of control points is called the *tautological projection* and written $\pi_{\mathscr{A}}$. Restricting the tautological projection to $X_{\mathscr{A}}$ gives the *algebraic moment map* $\mu\colon X_{\mathscr{A}} \to \Delta$. The components of its inverse $\mu^{-1}\colon \Delta \xrightarrow{\sim} X_{\mathscr{A}}$ provide a preferred set of blending functions for the patch. When $\mathscr{A} \subset \mathbb{Z}^d$, these were studied in [7, 6], where they were shown to have linear precision, and that they may be computed by *iterative proportional fitting (IPF)*, a numerical algorithm from statistics [5]. These same arguments apply to irrational toric patches—the preferred blending functions have linear precision and are computed by IPF.

Any patch has unique blending functions with linear precision [7, Theorem 1.11]. While the classification of toric patches for which these preferred blending functions are rational functions remains open in general, it has been settled for surface patches ($d = 2$) [13], and this places very strong restrictions on higher-dimensional patches.

2.2 Iterative Proportional Fitting for Toric Patches

In algebraic statistics, $\triangle^{\mathscr{A}}$ is identified with the probability simplex parameterizing probability distributions on data indexed by $\mathscr{A}$. The image $X_{\mathscr{A},w} := w.X_{\mathscr{A}}$ of $\mathbb{R}^d_>$ under the map $\varphi_{\mathscr{A}}$ (7) and translation by w (5) is known as a *toric model* [14, §1.2]. It is more common to call this a *log-linear model*, as the logarithms of the coordinates of $\varphi_{\mathscr{A}}$ are linear functions in the logarithms of the coordinates of $\mathbb{R}^d_>$, or a *discrete exponential family* as the coordinates of $\varphi_{\mathscr{A}}$ are exponentials in the logarithms of the coordinates of $\mathbb{R}^d_>$.

The tautological map appears in statistics as follows. Given (observed) normalized data $q \in \triangle^{\mathscr{A}}$, the problem of *maximum likelihood estimation* asks for a probability distribution in the toric model, $p \in X_{\mathscr{A},w}$, with the same sufficient statistics as q, $\mu(p) = \pi_{\mathscr{A}}(p) = \pi_{\mathscr{A}}(q)$. By Birch's Theorem, the point $p \in X_{\mathscr{A},w}$ is unique and hence

$$p \;=\; \mu^{-1}(\pi_{\mathscr{A}}(q)) \;\;.$$

Thus inverting the tautological projection is necessary for maximum likelihood estimation.

Darroch and Ratcliff [5] introduced the numerical algorithm of iterative proportional fitting, also known as generalized iterative scaling, for computing the inverse μ^{-1} of the tautological projection. We now describe their algorithm.

Observe first that the toric patch $X_{\mathscr{A},w}$ does not change if we translate all elements of $\mathscr{A}$ by a fixed vector $\mathbf{b}$, $(\mathbf{a} \mapsto \mathbf{a}+\mathbf{b})$, so we may assume that $\mathscr{A}$ lies in the positive orthant $\mathbb{R}^d_>$. Scaling the exponent vectors in $\mathscr{A}$ by a fixed positive scalar $t \in \mathbb{R}_>$ also does not change $X_{\mathscr{A},w}$ as $x \mapsto x^t$ is a homeomorphism of $\mathbb{R}_>$ which extends to a homeomorphism of $\mathbb{R}^d_\geq$. Thus we may assume that $\mathscr{A}$ lies in the standard simplex Δ_d in $\mathbb{R}^d$,

$$\Delta_d \;=\; \{x \in \mathbb{R}^d_\geq \mid \textstyle\sum x \leq 1\} \;\;.$$

Lastly, we lift this to the probability simplex $\triangle^{d+1} \subset \mathbb{R}^{d+1}_\geq$,

$$\triangle^{d+1} \;:=\; \{y \in \mathbb{R}^{d+1}_\geq \mid \textstyle\sum y = 1\} \;\;,$$

by

$$\mathscr{A} \ni \mathbf{a} \longmapsto \mathbf{a}^+ := (1-\sum \mathbf{a}, \mathbf{a}) \in \Delta^{d+1} \ .$$

Since for $t \in \mathbb{R}_>$ and $x \in \mathbb{R}^d_>$,

$$(t,tx)^{\mathbf{a}^+} \ = \ t^{1-\sum \mathbf{a}}(tx)^{\mathbf{a}} \ = \ tx^{\mathbf{a}} \ ,$$

we see that replacing $\mathscr{A}$ by this homogeneous version also does not change $X_{\mathscr{A}}$.

We describe the algorithm of *iterative proportional fitting*, which is Theorem 1 in [5].

Proposition 6. *Suppose that $\mathscr{A} \subset \Delta^{d+1}$ has convex hull Δ and $q \in \Delta^{\mathscr{A}}$. Set $y := \pi_{\mathscr{A}}(q) \in \Delta$. Then the sequence of points*

$$\{p^{(m)} \mid m = 0,1,2\ldots\} \subset \Delta^{\mathscr{A}}$$

whose $\mathbf{a}$*-coordinates are defined by $p^{(0)}_{\mathbf{a}} := w_{\mathbf{a}}$ and, for $m \geq 0$,*

$$p^{(m+1)}_{\mathbf{a}} \ := \ p^{(m)}_{\mathbf{a}} \cdot \frac{y^{\mathbf{a}}}{(\pi_{\mathscr{A}}(p^{(m)}))^{\mathbf{a}}} \ ,$$

converges to the unique point $p \in X_{\mathscr{A},w}$ such that $\mu(p) = \pi_{\mathscr{A}}(p) = \pi_{\mathscr{A}}(q) = y$.

We remark that if $\mathscr{A}$ is not homogenized then to compute $\mu^{-1}(y)$ for $y \in \Delta$, we first put $\mathscr{A}$ into homogeneous form using an affine map ψ, and then use iterative proportional fitting to compute $\pi^{-1}_{\mathscr{A}^+}(\psi(y)) = \pi^{-1}_{\mathscr{A}}(y)$. We also call this modification of the algorithm of Proposition 6 iterative proportional fitting. Thus iterative proportional fitting computes the inverse image of the tautological projection.

3 Injectivity of Patches

Birch's Theorem (Theorem 4) states that for one particular choice of control points, namely $\{\mathbf{b}_{\mathbf{a}} := \mathbf{a} \mid \mathbf{a} \in \mathscr{A}\}$ and all weights $w \in \mathbb{R}^{\mathscr{A}}_>$, the mapping F (1) of a toric patch of shape $(\mathscr{A}, w)$ is a homeomorphism onto its image. From this, we can infer that for *most* choices of control points and weights, this mapping is at least an immersion. To study dynamical systems arising from chemical reaction networks, Craciun and Feinberg [2] prove an injectivity theorem for certain maps, which we adapt to generalize Birch's Theorem. This will give conditions on control points $\mathscr{B} \subset \mathbb{R}^d$ which guarantee that for *any* choice w of weights, the resulting mapping F (1) of a toric patch of shape $(\mathscr{A}, w)$ is a homeomorphism onto its image. This result has several consequences concerning the injectivity of toric patches.

Let us first give the Craciun-Feinberg Theorem. Let $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_m\} \subset \mathbb{R}^n$ be a finite set of points which affinely spans $\mathbb{R}^n$. For $k \in \mathbb{R}^m_>$ and $Z := \{\mathbf{z}_1, \ldots, \mathbf{z}_m\} \subset \mathbb{R}^n$, consider the map $G_k \colon \mathbb{R}^n_> \to \mathbb{R}^n$ defined by

$$G_k(x) \ := \ \sum_{i=1}^{m} k_i x^{\mathbf{y}_i} \mathbf{z}_i \ . \tag{8}$$

Theorem 7 (Craciun-Feinberg). *The map G_k is injective for every $k \in \mathbb{R}^m_>$ if and only if the determinant of the Jacobian matrix,*

$$\mathrm{Jac}(G_k) \;=\; \left(\frac{\partial (G_k)_i}{\partial x_j}\right)^n_{i,j=1} \quad ,$$

does not vanish for any $x \in \mathbb{R}^n_>$ and any $k \in \mathbb{R}^m_>$.

We give a proof in Appendix A.

The condition of Theorem 7 that the Jacobian $\mathrm{Jac}(G_k)$ does not vanish for any $x \in \mathbb{R}^n_>$ is reminiscent of the Jacobian conjecture [15], which is that the Jacobian of a polynomial map $G\colon \mathbb{C}^n \to \mathbb{C}^n$ does not vanish if and only if the map G is an isomorphism. Since we are restricted to $x \in \mathbb{R}^n_>$, it is closer to the real Jacobian conjecture, which is however false [16], and therefore not necessarily relevant.

The condition of Theorem 7 is conveniently restated in terms of Y and Z. For a list $I = \{i_1, \ldots, i_n\} \subset \{1, \ldots, m\}$, which we write as $I \in \binom{[m]}{n}$, let Y_I be the determinant of the matrix whose columns are the vectors $\mathbf{y}_{i_1}, \ldots, \mathbf{y}_{i_n}$, and define Z_I similarly. In Appendix A, we deduce the following corollary.

Corollary 8. *The map G_k (8) is injective for all $k \in \mathbb{R}^m_>$ if and only if $(Y_I \cdot Z_I) \cdot (Y_J \cdot Z_J) \geq 0$ for every $I, J \in \binom{[m]}{n}$ and at least one product $Y_I \cdot Z_I$ is non-zero.*

This leads to a generalization of Birch's Theorem. An ordered list $p_0, \ldots, p_d$ of affinely independent points in $\mathbb{R}^d$ determines an orientation of $\mathbb{R}^d$—simply consider the basis

$$p_1 - p_0\,,\; p_2 - p_0\,,\; \cdots\,,\; p_d - p_0 \quad .$$

Let $\mathscr{A}$ and $\mathscr{B} = \{\mathbf{b}_\mathbf{a} \mid \mathbf{a} \in \mathscr{A}\}$ be finite sets of points in $\mathbb{R}^d$. Suppose that $\{\mathbf{a}_0, \ldots, \mathbf{a}_d\}$ is an affinely independent subset of $\mathscr{A}$. If the corresponding subset $\{\mathbf{b}_{\mathbf{a}_0}, \ldots, \mathbf{b}_{\mathbf{a}_d}\}$ of $\mathscr{B}$ is also affinely independent, then each subset determines an orientation, and the two orientations are either the same or they are opposite. We say that $\mathscr{A}$ and $\mathscr{B}$ are *compatible* if either every such pair of orientations is the same, or if every such pair of orientations is opposite. We further need that there is at least one affinely independent subset of $\mathscr{A}$ such that the corresponding subset of $\mathscr{B}$ is also affinely independent. Observe that compatibility is preserved by invertible affine transformations acting separately on $\mathscr{A}$ and $\mathscr{B}$.

In Fig. 3 shows three sets of labeled points. The first and second sets are compatible, but neither is compatible with the third.

We give our generalization of Birch's Theorem. Suppose that $\Delta \subset \mathbb{R}^d$ is the convex hull of $\mathscr{A}$ and $\{\beta_\mathbf{a}\colon \Delta \to \mathbb{R}_\geq \mid \mathbf{a} \in \mathscr{A}\}$ are toric Bézier functions for $\mathscr{A}$. For any $w \in \mathbb{R}^\mathscr{A}_>$, let $F_w\colon \Delta \to \mathbb{R}^d$ be the toric patch of shape $(\mathscr{A}, w)$ given by the control points $\mathscr{B} \subset \mathbb{R}^d$:

$$F_w(x) \;:=\; \frac{\sum_{\mathbf{a}\in\mathscr{A}} w_\mathbf{a}\beta_\mathbf{a}(x)\,\mathbf{b}_\mathbf{a}}{\sum_{\mathbf{a}\in\mathscr{A}} w_\mathbf{a}\beta_\mathbf{a}(x)} \quad . \tag{9}$$

Theorem 9. *The map F_w is injective for all $w \in \mathbb{R}^\mathscr{A}_>$ if and only if $\mathscr{A}$ and $\mathscr{B}$ are compatible.*

As any set $\mathscr{A}$ is compatible with itself, this implies Birch's Theorem (Theorem 4).

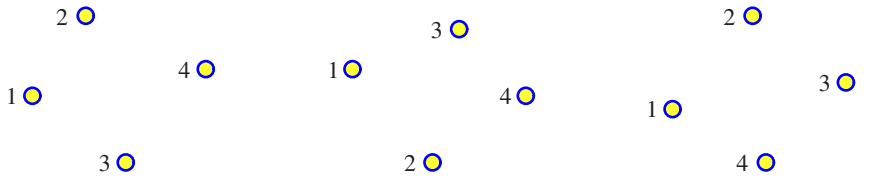


Fig. 3. Compatible and incompatible sets of points

Example 10. Let $\triangle$ be the convex hull of $\{(0,0),(1,0),(0,1)\}$. Set $\mathscr{A} := 3\triangle \cap \mathbb{Z}^2$ and let $w \in \mathbb{R}_{>}^{\mathscr{A}}$ be the weights of a cubic Bézier patch (Example 2 with $m = 3$). We consider choices $\mathscr{B} \subset \mathbb{R}^2$ of control points that are compatible with $\mathscr{A}$. For convenience, we will require that $\mathbf{b_a} = \mathbf{a}$ when $\mathbf{a}$ is a vertex and that if $\mathbf{a}$ lies on an edge of $3\triangle$, then so does $\mathbf{b_a}$. For these edge control points, compatibility imposes the restriction that they appear along the edge in the same order as the corresponding exponents from $\mathscr{A}$. The placement of the center control point is however constrained. We show two compatible choices of $\mathscr{B}$ in Fig. 4. On the left is the situation of Birch's Theorem, in which $\mathbf{b_a} = \mathbf{a}$, and on the right we have moved the edge control points. The region in which we are free to move the center point is shaded in each picture.

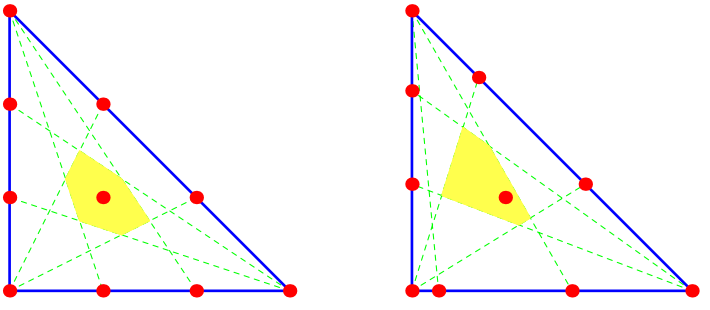


Fig. 4. Compatible control points for the Bézier triangle

Proof (Theorem 9). Let (t,x) be coordinates for $\mathbb{R}^{d+1}$ and consider the map $G_w\colon \mathbb{R}_{>}^{d+1} \to \mathbb{R}^{d+1}$ defined by

$$G_w(t,x) \;=\; \sum_{\mathbf{a}\in\mathscr{A}} t x^{\mathbf{a}} w_{\mathbf{a}}(1,\mathbf{b_a}) \;\;.$$

We claim that F_w is injective if and only if G_w is injective.

Since F_w is the composition (6) $\Delta \xrightarrow{\beta} X_{\mathscr{A}} \xrightarrow{w.} w.X_{\mathscr{A}} \xrightarrow{\pi_{\mathscr{B}}} \mathbb{R}^d$, with the first map an isomorphism, F_w is injective if and only if the composition of the last two maps is injective. Since $X_{\mathscr{A}}$ is compact, this will be injective if and only if its restriction to the

interior $X^\circ_{\mathscr{A}}$ of $X_{\mathscr{A}}$ is injective. Precomposing with the monomial parametrization (7) of $X^\circ_{\mathscr{A}}$, we see that F_w is injective if and only if the map $H_w\colon \mathbb{R}^d_> \to \mathbb{R}^d$ defined by

$$H_w(x) \;=\; \frac{\sum_{\mathbf{a}\in\mathscr{A}} x^{\mathbf{a}} w_{\mathbf{a}} \mathbf{b}_{\mathbf{a}}}{\sum_{\mathbf{a}\in\mathscr{A}} x^{\mathbf{a}} w_{\mathbf{a}}} \;:\; \mathbb{R}^d_> \xrightarrow{\varphi_{\mathscr{A}}} X_{\mathscr{A}} \xrightarrow{w.} w.X_{\mathscr{A}} \xrightarrow{\pi_{\mathscr{B}}} \mathbb{R}^d$$

is injective.

Since $G_w(t,x) = t\cdot G_w(1,x)$, these values lie on a ray through the origin, and we invite the reader to check that this ray meets the hyperplane with first coordinate 1 at the point $(1, H_w(x))$. Thus G_w is injective if and only if H_w is injective, which is equivalent to the map F_w being injective.

We deduce the theorem by showing that G_w is injective. This follows from Corollary 8 as the condition that $\mathscr{A}$ and $\mathscr{B}$ are compatible is equivalent to $Y_I \cdot Z_I \geq 0$ for all $I \in \binom{[m]}{d+1}$, where $Y = \{(1,\mathbf{a}) \mid \mathbf{a}\in\mathscr{A}\}$, $Z = \{(1,\mathbf{b}_{\mathbf{a}}) \mid \mathbf{a}\in\mathscr{A}\}$ and we have $m = \#\mathscr{A}$. □

We now describe two applications of Theorem 9 to modeling.

3.1 Solid Modeling with (Toric) Bézier Patches

In solid modeling, we represent a 3-dimensional solid by covering it with 3-dimensional patches, for example using Bézier toric patches as finite elements. Besides the obvious C^0 or higher continuity along the boundary as required, such Bézier finite elements should at least provide a one-to-one parametrization of their image, i.e. they should be injective. By Theorem 9, we may guarantee injectivity by requiring that the control points $\mathscr{B}$ be compatible with the exponents $\mathscr{A}$. Moreover, if these sets are incompatible, then there is some choice of weights for which the patch is not injective.

3.2 Injectivity of Bézier Curves and Surfaces

Typically, the exponents $\mathscr{A}$ and the control points do not lie in the same space; surfaces ($\mathscr{A}\subset\mathbb{R}^2$) are modeled in 3-space ($\mathscr{B}\subset\mathbb{R}^3$), or curves ($\mathscr{A}\subset\mathbb{R}$) in 2– or 3–space ($\mathscr{B}\subset\mathbb{R}^2$ or $\mathbb{R}^3$). Nevertheless, Theorem 9 gives conditions that imply injectivity of patches.

Let $p\in\mathbb{R}^{n+1}$ be a point disjoint from a hyperplane, H. The projection $\mathbb{R}^{n+1} \dashrightarrow H$ with center p is the map which associates a point $x\in\mathbb{R}^{n+1}$ to the intersection of the line $\overline{px}$ with H. We use a broken arrow as the projection is not defined on the plane through p parallel to H. Identifying H with $\mathbb{R}^n$ gives a projection map $\mathbb{R}^{n+1} \dashrightarrow \mathbb{R}^n$. A coordinate projection $(x_1,\ldots,x_n,x_{n+1}) \mapsto (x_1,\ldots,x_n)$ is a projection with center at infinity. More generally, a *projection* $\mathbb{R}^n \dashrightarrow \mathbb{R}^d$ is a sequence of such projections from points.

Theorem 11. *Let $\mathscr{A}\subset\mathbb{R}^d$, $w\in\mathbb{R}^{\mathscr{A}}_>$, and $\mathscr{B}\subset\mathbb{R}^n$ be the exponents, weights, and control points of a toric patch and let Δ be the convex hull of $\mathscr{A}$. If there is a projection $\pi\colon \mathbb{R}^n \dashrightarrow \mathbb{R}^d$ such that $\mathscr{A}$ is compatible with the image $\pi(\mathscr{B})$ of $\mathscr{B}$, then the mapping $F\colon \Delta \to \mathbb{R}^n$ given by the toric blending functions associated to $\mathscr{A}$, the weights w, and control points $\mathscr{B}$ is injective.*

Proof. By Theorem 9, the composition $\pi \circ F$ is injective, from which it follows that F must have been injective. □

Example 12. For the curve on the left in Fig. 1 (which is reproduced in Fig. 5), the vertical projection $\mathbb{R}^2 \to \mathbb{R}^1$ maps the control points $\{\mathbf{b}_0, \ldots, \mathbf{b}_5\}$ to points on the line in the same order as the exponents $\mathscr{A} = \{0,1,2,3,4,5\}$, which implies that the curve has no self-intersections.

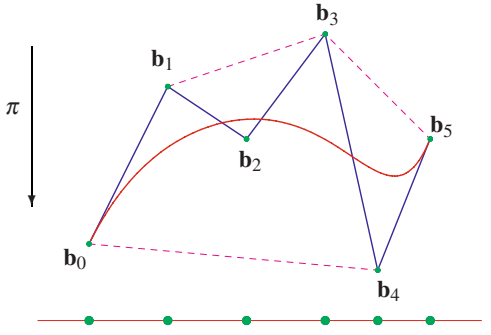


Fig. 5. A compatible projection

4 Control Polytopes and Toric Degenerations

The convex hull property asserts that the image, $F(\Delta)$, of a toric Bézier patch of shape $(\mathscr{A}, w)$ given by control points $\mathscr{B} = \{\mathbf{b}_\mathbf{a} \mid \mathbf{a} \in \mathscr{A}\} \subset \mathbb{R}^n$ and weights $w \in \mathbb{R}^{\mathscr{A}}_{>}$ lies in the convex hull of the control points. When $F(\Delta)$ is a curve, the control points may be joined sequentially to form the *control polygon*, which is a piecewise linear representation of the curve. When $F(\Delta)$ is however a surface patch, there are many ways to interpolate the control points by triangles or other polygons to obtain a piecewise linear surface, called a *control polytope*, that represents the patch. The shape of this control polytope affects the shape of the patch. For example, when the control points have the form $(\mathbf{a}, \lambda(\mathbf{a}))$ for λ a convex function, then the patch is convex [17, 1]. Also, Leroy [18] uses a particular control polytope for the graph of a function to obtain certificates of positivity for polynomials.

Among all control polytopes for a given set of control points, we identify the class of regular control polytopes, which come from regular triangulations of the exponents $\mathscr{A}$. These regular control polytopes are related to the shape of the patch in the following precise manner: There is a choice of weights so that a toric Bézier patch is arbitrarily close to a given control polytope if and only if that polytope is regular.

4.1 Bézier Curves

It is instructive to begin with Bézier curves. A Bézier curve of degree m in $\mathbb{R}^n$ with weights w is the composition (6),

$$[0,1] \xrightarrow{\beta} \Delta^{m+1} \xrightarrow{w.} \Delta^{m+1} \xrightarrow{\pi_{\mathscr{B}}} \mathbb{R}^n \ ,$$

where $\beta = (\beta_0, \ldots, \beta_m)$ with $\beta_i(x) = \binom{m}{i} x^i (1-x)^{m-i}$ for $x \in [0,1]$. Then the map β is given by $z_i = \binom{m}{i} x^i (1-x)^{m-i}$, for $i = 0, \ldots, m$. Here, $(z_0, \ldots, z_m) \in \mathbb{R}^{d+1}_{\geq}$ with $z_0 + \cdots + z_m = 1$ are the coordinates for Δ^{m+1}. The image $\beta[0,1] \subset \Delta^{m+1}$ is defined by the binomials

$$\tbinom{m}{i}\tbinom{m}{j} z_a z_b \ - \ \tbinom{m}{a}\tbinom{m}{b} z_i z_j \ = \ 0 , \qquad \text{for} \quad a+b = i+j \ . \tag{10}$$

To see this, suppose that $(z_0, \ldots, z_m) \in \mathbb{R}^{m+1}_{\geq}$ satisfies (10). Setting $x := z_1/(mz_0 + z_1)$, then we may solve these equations to obtain $z_i = \binom{m}{i} x^i (1-x)^{m-i} = \beta_i(x)$.

If $w = (w_0, \ldots, w_m) \in \mathbb{R}^{m+1}_{>}$ are weights, then $w.\beta[0,1]$ is defined in Δ^{m+1} by

$$w_i w_j \tbinom{m}{i}\tbinom{m}{j} z_a z_b \ - \ w_a w_b \tbinom{m}{a}\tbinom{m}{b} z_i z_j \ = \ 0 , \qquad \text{for} \quad a+b = i+j \ . \tag{11}$$

Suppose that we choose weights $w_i := t^{i(m-i)}$. Dividing by $t^{i(m-i)+j(m-j)}$, (11) becomes

$$\tbinom{m}{i}\tbinom{m}{j} z_a z_b \ - \ t^{i^2+j^2-a^2-b^2} \tbinom{m}{a}\tbinom{m}{b} z_i z_j \ = \ 0 , \qquad \text{for} \quad a+b = i+j \ . \tag{12}$$

Since $i+j = a+b$, we may assume that $a < i \leq j < b$. Setting $c := i-a = b-j \geq 1$, we see that

$$i^2 + j^2 - a^2 - b^2 \ = \ c(i+a) - c(j+b) \ = \ c(i-j+a-b) \leq -2c \ < \ 0 \ .$$

If we consider the limit of these binomials (12) as $t \to \infty$, we obtain

$$z_a \cdot z_b \ = \ 0 \qquad \text{if} \quad |a-b| > 1 \ .$$

These define the polygonal path in Δ^{m+1} whose ith segment is the edge

$$(\underbrace{0, \ldots, 0}_{i-1}, x, 1-x, \underbrace{0, \ldots, 0}_{m-i}) \qquad \text{for} \quad x \in [0,1] \ ,$$

and whose projection to $\mathbb{R}^n$ is the *control polygon* of the Bézier curve, which is the collection of line segments $\overline{\mathbf{b}_0, \mathbf{b}_1}$, $\overline{\mathbf{b}_1, \mathbf{b}_2}$, …, $\overline{\mathbf{b}_{m-1}, \mathbf{b}_m}$.

We illustrate this when $m = 3$ in Fig. 6, which shows three different Bézier curves having the same control points, but different weights $w_i = t^{i(3-i)}$ for $t = 1, 3, 9$. In algebraic geometry, altering the weights in this manner is called a *toric degeneration*. The Bézier cubics are displayed together with the cubics $w.\beta[0,1]$ lying in the 3-simplex, Δ^4, which is drawn in $\mathbb{R}^3$. In these pictures, the projection $\pi_{\mathscr{B}}$ is simply the vertical projection forgetting the third coordinate. The progression indicated in Fig. 6, where the Bézier curve approaches the control polygon as the parameter t increases, is a general phenomenon. Let $\|\cdot\|$ be the usual Euclidean distance in $\mathbb{R}^n$.

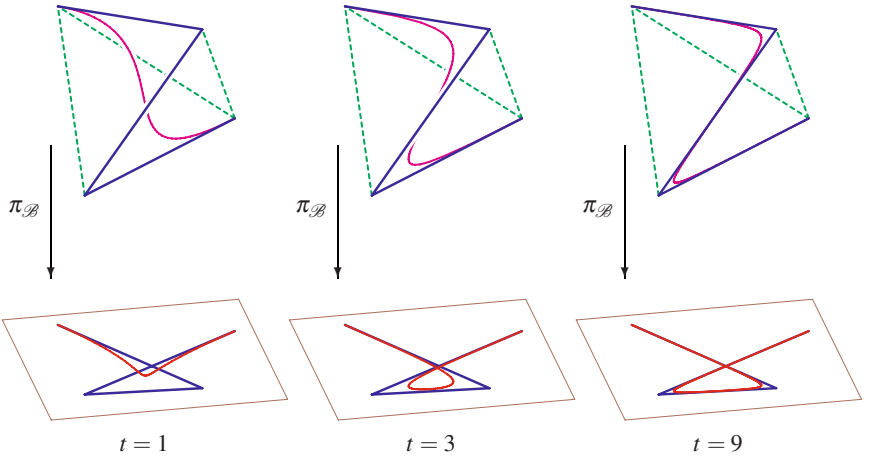


Fig. 6. Toric degenerations of a Bézier cubic

Theorem 13. *Suppose that $F_t\colon [0,1] \to \mathbb{R}^n$ is a Bézier curve of degree m with control points $\mathscr{B}$ and weights $w_i = t^{i(m-i)}$. Set $\kappa := \max\{\|\mathbf{b_a}\| : \mathbf{b_a} \in \mathscr{B}\}$. For any $\varepsilon > 0$, if we have $t > \kappa m/\varepsilon$, then the distance between the control polygon and any point of the Bézier curve $F_t[0,1]$ is less than ε.*

Proof. Let $z \in w.\beta[0,1] \subset \Delta^{m+1}$, where the weights are $w_i = t^{i(m-i)}$ with $t > \kappa m/\varepsilon$. Suppose that $b-a > 1$ are integers in $[0,m]$. Then there exist integers $i \le j$ with $a < i \le j < b$ and $a+b = i+j$. By (12), we have

$$z_a z_b \;=\; t^{i^2+j^2-a^2-b^2}\,\frac{\binom{m}{a}\binom{m}{b}}{\binom{m}{i}\binom{m}{j}} z_i z_j \quad .$$

Since the binomial coefficients are log-concave, we have

$$\binom{m}{a}\binom{m}{b} \;<\; \binom{m}{i}\binom{m}{j} \quad .$$

Using $i^2+j^2-a^2-b^2 < -2$ and $z_i + z_j \le 1$, we see that

$$z_a z_b \;<\; \frac{1}{4t^2} \quad .$$

In particular, if $|b-a| > 1$, then at most one of z_a or z_b exceeds $1/2t$. We conclude that at most two, necessarily consecutive, coordinates of z may exceed $1/2t$. Suppose that i is an index such that $z_j < 1/2t$ if $j \ne i-1, i$ and let $x := z_{i-1}\mathbf{b}_{i-1} + (1-z_{i-1})\mathbf{b}_i$, a point along the ith segment of the control polygon. Since

$$1 \;\ge\; z_{i-1} + z_i \;=\; 1 - \sum_{j \ne i-1,i} z_j \;>\; 1 - \frac{m-1}{2t} \quad ,$$

we have $0 \leq 1 - z_{i-1} - z_i < \frac{m-1}{2t} < \frac{m}{2t}$, and we see that

$$\begin{aligned}\|\pi_{\mathscr{B}}(z) - x\| &= \Big\|\sum z_j \mathbf{b}_j \;-\; (z_{i-1}\mathbf{b}_{i-1} + (1-z_{i-1})\mathbf{b}_i)\Big\| \\ &\leq \sum_{j\neq i-1,i} z_j\|\mathbf{b}_j\| \;+\; |z_i - (1-z_{i-1})|\|\mathbf{b}_i\| \\ &< \kappa\frac{m-1}{2t} \;+\; \kappa\frac{m}{2t} \;<\; \frac{\kappa m}{t} \;=\; \varepsilon\,.\end{aligned}$$

This proves the theorem as $z \in w.\beta[0,1]$ is an arbitrary point of the curve $F[0,1]$. □

4.2 Regular Triangulations and Control Polytopes

In dimensions 2 and higher, the analog of Theorem 13 requires the notion of a regular triangulation from geometric combinatorics. Let $\mathscr{A} \subset \mathbb{R}^d$ be a finite set of points and consider a lifting function $\lambda : \mathscr{A} \to \mathbb{R}$. Let P_λ be the convex hull of the lifted points

$$\lambda(\mathscr{A}) \;:=\; \{(\mathbf{a}, \lambda(\mathbf{a})) \mid \mathbf{a} \in \mathscr{A}\} \;\subset\; \mathbb{R}^{d+1}\quad.$$

We assume that P_λ is full-dimensional in that $\mathbb{R}^{d+1}$ is its affine span.

The *upper facets* of P_λ are those facets whose outward pointing normal vector has positive last coordinate. Any face of an upper facet is an *upper face*. We illustrate this below when $d = 1$, where the displayed arrows are outward pointing normal vectors to upper facets.

Projecting these upper facets to $\mathbb{R}^d$ yields a *regular polyhedral subdivision* of the convex hull of $\mathscr{A}$, which is the image of P_λ. For our purposes, we will need to assume that the lifting function is generic in that all upper facets of P_λ are simplices. In this case, we obtain a *regular triangulation* of $\mathscr{A}$. This consists of a collection

$$\{\mathscr{A}_i \mid i = 1, \ldots, m\}$$

of subsets of $\mathscr{A}$, where each subset $\mathscr{A}_i$ consists of $d+1$ elements and spans a d-dimensional simplex. We regard all subsets of the facets $\mathscr{A}_i$ as faces of the triangulation. These simplices form a subdivision in that they cover the convex hull of $\mathscr{A}$ and any two with a non-empty intersection meet along a common face.

The subdivision induced by the lifting function in Fig. 7 consists of three intervals resulting from removal of the middle point of $\mathscr{A}$, which does not participate in the subdivision as it is not lifted high enough.

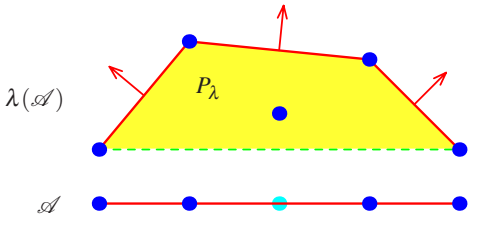


Fig. 7. Upper faces and a regular subdivision

A set may have many regular triangulations, and not every point needs to participate in a given triangulation. Fig. 8 shows the edges in four regular triangulations of $3\triangle \cap \mathbb{Z}^2$.

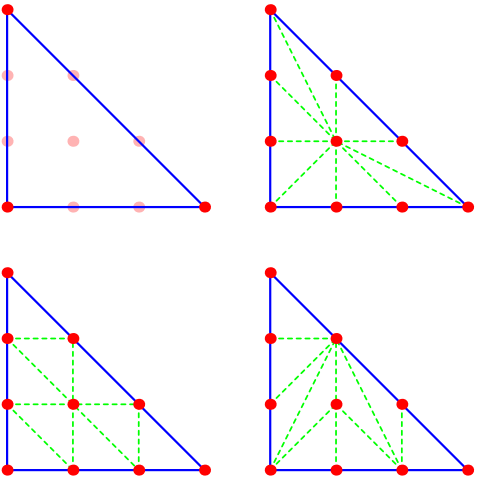


Fig. 8. Some regular triangulations

Not every triangulation is regular. We may assume that a lifting function λ for the triangulation of $4\triangle \cap \mathbb{Z}^2$ in Fig. 9 takes a constant value at the three interior points. The clockwise neighbor of any vertex of the big triangle must be lifted lower than that vertex. (Consider the figure they form with the parallel edge of the interior triangle.) Since the edge of the big triangle is lifted to a convex path, this is impossible, except in some M.C. Escher woodcuts.

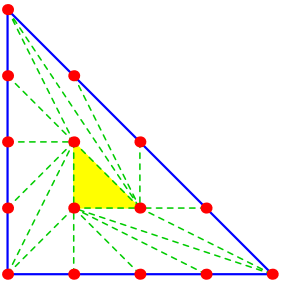


Fig. 9. An irregular triangulation

Definition 14. *Let $\mathscr{B} = \{\mathbf{b}_\mathbf{a} \mid \mathbf{a} \in \mathscr{A}\} \subset \mathbb{R}^n$ be a collection of control points indexed by a finite set of exponents $\mathscr{A} \subset \mathbb{R}^d$ with $d \leq n$. Given a regular triangulation $\mathscr{T} = \{\mathscr{A}_i \mid i = 1, \ldots, m\}$ of $\mathscr{A}$ we define the control polytope as follows. For each d-simplex $\mathscr{A}_i$ in $\mathscr{T}$, the corresponding points of $\mathscr{B}$ span a (possibly degenerate) simplex*

$$conv\{\mathbf{b}_\mathbf{a} \mid \mathbf{a} \in \mathscr{A}_i\} \quad .$$

The union of these simplices in $\mathbb{R}^n$ forms the regular control polytope $\mathscr{B}(\mathscr{T})$ that is induced by the regular triangulation $\mathscr{T}$ of $\mathscr{A}$. This is a simplicial complex in $\mathbb{R}^n$ with vertices in $\mathscr{B}$ that has the same combinatorial type as the triangulation $\mathscr{T}$ of $\mathscr{A}$.

If the coordinate points $(\mathbf{e}_\mathbf{a} \mid \mathbf{a} \in \mathscr{A})$ of $\mathbb{R}^\mathscr{A}$ are our control points (these are the vertices of $\triangle^\mathscr{A}$), then the regular control polytope is just the geometric realization $|\mathscr{T}|$ of the simplicial complex $\mathscr{T}$, which is a subcomplex of the simplex $\triangle^\mathscr{A}$. In general, $\mathscr{B}(\mathscr{T})$ is the image of this geometric realization $|\mathscr{T}| \subset \triangle^\mathscr{A}$ under the projection $\pi_\mathscr{B}$.

Example 15. Let $\mathscr{A} := 3\triangle \cap \mathbb{Z}^2$, the exponents for a cubic Bézier triangle. Fig. 10 shows the three control polytopes corresponding to the last three regular triangulations of Fig. 8, all with the same control points.

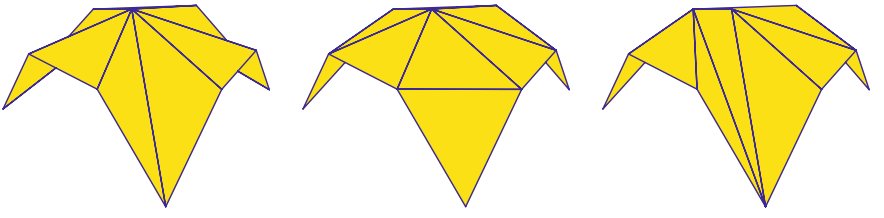


Fig. 10. Three control polytopes

The reason that we introduce regular control polytopes is that they may be approximated by toric Bézier patches.

Theorem 16. *Let $\mathscr{A} \subset \mathbb{R}^d$, $w \in \mathbb{R}^\mathscr{A}_>$, and $\mathscr{B} \subset \mathbb{R}^n$ be exponents, weights, and control points for a toric Bézier patch. Suppose that $\mathscr{T}$ is a regular triangulation of $\mathscr{A}$ induced by a lifting function $\lambda : \mathscr{A} \to \mathbb{R}$. For each $t > 0$, let $F_t : \Delta \to \mathbb{R}^n$ be the toric Bézier patch of shape $\mathscr{A}$ with control points $\mathscr{B}$ and weights $t^{\lambda(\mathbf{a})} w_\mathbf{a}$. Then, for any $\varepsilon > 0$ there exists a t_0 such that if $t > t_0$, the image $F_t(\Delta)$ lies within ε of the control polytope $\mathscr{B}(\mathscr{T})$.*

We prove Theorem 16 in Appendix B. Fig. 11 illustrates Theorem 16 for a cubic Bézier triangle with the control points of Example 15. The patch on the left is the cubic Bézier triangle with the weights of Example 2. The second and third patches are its deformations corresponding to the lifting function inducing the leftmost control polytope of Fig. 10. The values of t are 1, 5, and 100, as we move from left to right.

An absolutely unpractical consequence of Theorem 16 is a universality result: Any surface which admits a triangulation that forms a regular control polytope may be approximated by a single Bézier patch.

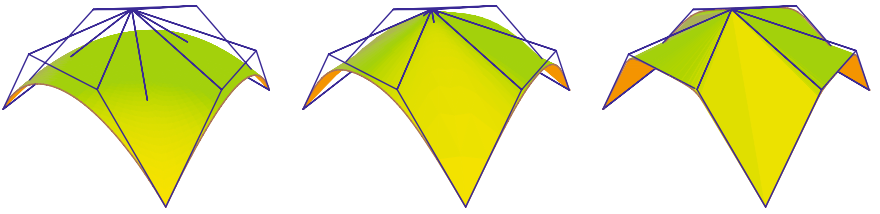


Fig. 11. Degeneration to the control polytope

As with Theorem 13, the main idea behind the proof of Theorem 16 (which is given in Appendix B) is that for t large enough, the translated patch $t.X_{\mathscr{A},w} \subset \Delta^{\mathscr{A}}$ can be made arbitrarily close to the geometric realization $|\mathscr{T}| \subset \Delta^{\mathscr{A}}$ of the regular triangulation $\mathscr{T}$. The result follows by projecting this into $\mathbb{R}^n$ using $\pi_{\mathscr{B}}$.

In Appendix B we also prove a weak converse to Theorem 16. Namely if $w.X_{\mathscr{A}}$ is sufficiently close to the geometric realization $|\mathscr{T}|$ of a triangulation $\mathscr{T}$, then $\mathscr{T}$ is in fact the regular triangulation of $\mathscr{A}$ induced by the lifting function $\lambda(\mathbf{a}) = \log(w_{\mathbf{a}})$.

Theorem 17. *Let $\mathscr{A} \subset \mathbb{R}^d$ be a finite set of exponents. Suppose that $|\mathscr{T}| \subset \Delta^{\mathscr{A}}$ is the geometric realization of a triangulation $\mathscr{T}$ of $\mathscr{A}$ and there is a weight w such that the distance between $X_{\mathscr{A},w}$ and $|\mathscr{T}|$ is less than $1/2(d+1)$. Then $\mathscr{T}$ is the regular triangulation induced by the lifting function $\lambda(\mathbf{a}) = \log(w_{\mathbf{a}})$.*

References

1. Dahmen, W.: Convexity and Bernstein-Bézier polynomials. In: Curves and surfaces (Chamonix-Mont-Blanc, 1990), pp. 107–134. Academic Press, Boston (1991)
2. Craciun, G., Feinberg, M.: Multiple equilibria in complex chemical reaction networks. I. The injectivity property. SIAM J. Appl. Math. 65(5), 1526–1546 (2005)
3. Krasauskas, R.: Toric surface patches. Adv. Comput. Math. 17(1-2), 89–133 (2002); Advances in geometrical algorithms and representations
4. Brown, L.D.: Fundamentals of statistical exponential families with applications in statistical decision theory. Institute of Mathematical Statistics Lecture Notes—Monograph Series, 9. Institute of Mathematical Statistics, Hayward, CA (1986)
5. Darroch, J.N., Ratcliff, D.: Generalized iterative scaling for log-linear models. Ann. Math. Statist. 43, 1470–1480 (1972)
6. Sottile, F.: Toric ideals, real toric varieties, and the moment map. In: Topics in algebraic geometry and geometric modeling. Contemp. Math., vol. 334, pp. 225–240. Amer. Math. Soc., Providence (2003)
7. Garcia-Puente, L.D., Sottile, F.: Linear precision for parametric patches. Advances in Computational Mathematics (to appear, 2009)
8. Karčiauskas, K., Krasauskas, R.: Comparison of different multisided patches using algebraic geometry. In: Laurent, P.J., Sablonniere, P., Schumaker, L. (eds.) Curve and Surface Design: Saint-Malo 1999, pp. 163–172. Vanderbilt University Press, Nashville (2000)
9. Cox, D.: What is a toric variety? In: Topics in algebraic geometry and geometric modeling. Contemp. Math., vol. 334, pp. 203–223. Amer. Math. Soc., Providence (2003)

10. DeRose, T., Goldman, R., Hagen, H., Mann, S.: Functional composition algorithms via blossoming. ACM Trans. on Graphics 12, 113–135 (1993)
11. Fulton, W.: Introduction to toric varieties. Annals of Mathematics Studies, vol. 131. Princeton University Press, Princeton (1993); The William H. Roever Lectures in Geometry
12. Agresti, A.: Categorical Data Analysis. Wiley series in Probability and Mathematical Statistics. Wiley, New York (1990)
13. Graf von Bothmer, H.C., Ranestad, K., Sottile, F.: Linear precision for toric surface patches, ArXiv:math/0806.3230 (2007)
14. Pachter, L., Sturmfels, B. (eds.): Algebraic statistics for computational biology. Cambridge University Press, New York (2005)
15. Keller, O.: Ganze cremonatransformationen. Monatschr. Math. Phys. 47, 229–306 (1939)
16. Pinchuk, S.: A counterexample to the strong real Jacobian conjecture. Math. Z. 217(1), 1–4 (1994)
17. Dahmen, W., Micchelli, C.A.: Convexity of multivariate Bernstein polynomials and box spline surfaces. Studia Sci. Math. Hungar. 23(1-2), 265–287 (1988)
18. Leroy, R.: Certificats de positivité et minimisation polynomiale dans la base de Bernstein multivariée. PhD thesis, Institut de Recherche Mathématique de Rennes (2008)
19. Sturmfels, B.: Gröbner bases and convex polytopes. American Mathematical Society, Providence (1996)

Appendix

A Proofs of Injectivity Theorems

Theorem 7. (Craciun-Feinberg) *The map G_k is injective for every $k \in \mathbb{R}^m_>$ if and only if the determinant of the Jacobian matrix,*

$$\mathrm{Jac}(G_k) \;=\; \left(\frac{\partial (G_k)_i}{\partial x_j}\right)^n_{i,j=1} \quad ,$$

does not vanish for any $x \in \mathbb{R}^n_>$ and any $k \in \mathbb{R}^m_>$.

Proof. We show the equivalence of the two statements, transforming one into the other. First, suppose there is a $k \in \mathbb{R}^m_>$ so that G_k is not injective. Then there exist $c, d \in \mathbb{R}^n_>$ so that $G_k(c) = G_k(d)$. Then we have

$$0 \;=\; \sum_{i=1}^m k_i(c^{\mathbf{y}_i} - d^{\mathbf{y}_i})\mathbf{z}_i \;=\; \sum_{i=1}^m k'_i((\tfrac{c}{d})^{\mathbf{y}_i} - 1)\mathbf{z}_i \quad ,$$

where $k' \in \mathbb{R}^m_>$ is defined by $k'_i = k_i d^{\mathbf{y}_i}$, and $\frac{c}{d} \in \mathbb{R}^n_>$ has ith coordinate $\frac{c_i}{d_i}$. In particular, $G_{k'}(\frac{c}{d}) = G_{k'}(\iota)$, where $\iota := (1, \ldots, 1)$. Define $\mathbf{v} \in \mathbb{R}^n$ by

$$v_i \;:=\; \log(c_i) - \log(d_i) \quad ,$$

so that $e^{\mathbf{y}_i \cdot \mathbf{v}} = (\frac{c}{d})^{\mathbf{y}_i}$, where $\mathbf{y}_i \cdot \mathbf{v}$ is the Euclidean dot product, and we now have

$$0 \;=\; \sum_{i=1}^m k'_i(e^{\mathbf{y}_i \cdot \mathbf{v}} - 1)\mathbf{z}_i \quad . \tag{13}$$

Define the univariate function by $f(t) = (e^t - 1)/t$ for $t \neq 0$ and set $f(0) = e$. Then f is an increasing continuous bijection between $\mathbb{R}$ and $\mathbb{R}_>$. Define $k'' \in \mathbb{R}^m_>$ by $k''_i := k'_i f(\mathbf{y}_i \cdot \mathbf{v})$. Then $k'_i(e^{\mathbf{y}_i \cdot \mathbf{v}} - 1) = k''_i(\mathbf{y}_i \cdot \mathbf{v})$, and (13) becomes

$$0 \;=\; \sum_{i=1}^{m} k''_i(\mathbf{y}_i \cdot \mathbf{v})\mathbf{z}_i \;\; . \tag{14}$$

We claim that $\mathbf{v}$ lies in the kernel of the Jacobian matrix of $G_{k''}$ evaluated at the point ι. Indeed, let $\mathbf{e}_j := (0,\ldots,1,\ldots,0) \in \mathbb{R}^n$ be the unit vector in the jth direction. Then

$$\frac{\partial G_{k''}}{\partial x_j}(x) \;=\; \sum_{i=1}^{m} k''_i x^{\mathbf{y}_i - \mathbf{e}_j} y_{i,j}\mathbf{z}_i \;\; ,$$

where $\mathbf{y}_i = (y_{i,1},\ldots,y_{i,n})$. Since $\iota^{\mathbf{y}_i - \mathbf{e}_j} = 1$ and $\sum_j y_i v_j = \mathbf{y}_i \cdot \mathbf{v}$, we see that

$$\sum_{j=1}^{n} \frac{\partial G_{k''}}{\partial x_j}(\iota)\, v_j \;=\; \sum_{j=1}^{n}\sum_{i=1}^{m} k''_i \iota^{\mathbf{y}_i - \mathbf{e}_j} y_{i,j} v_j \mathbf{z}_i \;=\; \sum_{i=1}^{m} k''_i(\mathbf{y}_i \cdot \mathbf{v})\mathbf{z}_i \;=\; 0 \;\; ,$$

so that $\mathbf{v}$ lies in the kernel of the Jacobian matrix of $G_{k''}$ evaluated at ι, which implies that the Jacobian determinant of $G_{k''}$ vanishes at ι.

The theorem follows as these arguments are reversible. □

Corollary 8. *The map G_k (8) is injective for all $k \in \mathbb{R}^m_>$ if and only if $(Y_I \cdot Z_I)\cdot(Y_J \cdot Z_J) \geq 0$ for every $I,J \in \binom{[m]}{n}$ and at least one product $Y_I \cdot Z_I$ is non-zero.*

Proof. Observe first that the Jacobian matrix $\mathrm{Jac}(G_k)$ factors as the product of matrices $\delta^{-1} Y D Z^T$, where δ is the diagonal matrix with entries $(x_1,\ldots,x_n)$, D is the diagonal matrix with entries $(k_1 x^{\mathbf{y}_1},\ldots,k_m x^{\mathbf{y}_m})$ and Y and Z are the matrices whose columns are the vectors $\mathbf{y}_i$ and $\mathbf{z}_i$, respectively. If we apply the Binet-Cauchy Theorem to this factorization, we see that

$$\det \mathrm{Jac}(G_k) \;=\; x^{-\iota} \cdot \sum_{I \in \binom{[m]}{n}} \prod_{i \in I} k_i x^{\mathbf{y}_i} \cdot Y_I \cdot Z_I \;\; , \tag{15}$$

where $\iota = (1,\ldots,1)$.

Suppose that $(Y_I \cdot Z_I)\cdot(Y_J \cdot Z_J) \geq 0$ for every $I,J \in \binom{[m]}{n}$, and at least one product $Y_I \cdot Z_I$ is non-zero. Then all terms in the sum (15) have the same sign and not all are zero, and so the Jacobian does not vanish for any $x \in \mathbb{R}^n_>$ and $k \in \mathbb{R}^m_>$. Thus G_k is injective for all $k \in \mathbb{R}^m_>$, by Theorem 7.

Suppose that there are two subsets $I,J \in \binom{[m]}{n}$ such that $Y_I Z_I > 0$ and $Y_J Z_J < 0$. For $t \in \mathbb{R}_>$ and $K \in \binom{[m]}{n}$, define $k(K,t) \in \mathbb{R}^m_>$ by

$$k(K,t)_j \;:=\; \begin{cases} t & \text{if } j \in K \\ 1 & \text{otherwise} \end{cases}$$

If we fix $x \in \mathbb{R}^d_>$, then the expansion (15), implies that $\det \mathrm{Jac}(G_{k(K,t)})(x)$ has the same sign as $Y_K Z_K$ when $t \gg 0$, at least when $Y_K Z_K \neq 0$.

We conclude that there is some k,x such that $\det \mathrm{Jac}(G_k)(x) > 0$ and some k,x such that $\det \mathrm{Jac}(G_k)(x) < 0$, and therefore some k,x such that $\det \mathrm{Jac}(G_k)(x) = 0$. This implies the corollary. □

B Three Toric Theorems

Theorem 3. *Suppose that $\mathscr{A} \subset \mathbb{R}^d$ is a finite set of points with convex hull Δ. Let $\beta = \{\beta_{\mathbf{a}} \mid \mathbf{a} \in \mathscr{A}\}$ be a collection of irrational toric Bézier functions for $\mathscr{A}$. Then $\beta(\Delta) = X_{\mathscr{A}}$, the closure of the image of $\varphi_{\mathscr{A}}$.*

Proof. Let Δ° be the interior of Δ, which we assume has ℓ facets and is given by the facet inequalities $0 \leq h_i(x), i = 1, \ldots, \ell$. Define two maps $H\colon \Delta^\circ \to \mathbb{R}^\ell_>$ and $\psi\colon \mathbb{R}^\ell_> \to \mathbb{R}^{\mathscr{A}}$ by

$$\begin{aligned} H &: x \longmapsto (h_1(x), \ldots, h_\ell(x)), \\ \psi &: u \longmapsto [u_1^{h_1(\mathbf{a})} \cdots u_\ell^{h_\ell(\mathbf{a})} \,:\, \mathbf{a} \in \mathscr{A}]. \end{aligned}$$

Then the map $\beta\colon \Delta^\circ \to \mathbb{R}^{\mathscr{A}}$ (whose image is dense in $X_{\mathscr{A}}$) is the composition of the maps H and ψ.

$$\beta \,:\, \Delta^\circ \xrightarrow{\;H\;} \mathbb{R}^\ell \xrightarrow{\;\psi\;} \mathbb{R}^{\mathscr{A}} \quad .$$

Let us recall the definiton of the map $\varphi_{\mathscr{A}}\colon \mathbb{R}^d_> \to \mathbb{R}^{\mathscr{A}}$,

$$\varphi_{\mathscr{A}} \,:\, (x_1, \ldots, x_d) \longmapsto [x^{\mathbf{a}} \,:\, \mathbf{a} \in \mathscr{A}] \quad .$$

The theorem follows once we show that the map ψ factors through the map $\varphi_{\mathscr{A}}$. For this, define a new map $f_\Delta\colon \mathbb{R}^\ell_> \to \mathbb{R}^d_>$ by

$$f_\Delta \,:\, u \longmapsto t = (t_1, \ldots, t_d) \qquad \text{where} \quad t_j := \prod_{i=1}^{\ell} u_i^{\mathbf{v}_i \cdot \mathbf{e}_j} \quad .$$

Then we claim that

$$\psi \,:\, \mathbb{R}^\ell \xrightarrow{\;f_\Delta\;} \mathbb{R}^d_> \xrightarrow{\;\varphi_{\mathscr{A}}\;} \mathbb{R}^{\mathscr{A}} \quad .$$

To see this, we compute the component of $\psi(u)$ corresponding to $\mathbf{a} \in \mathscr{A}$,

$$\prod_{i=1}^{\ell} u_i^{h_i(\mathbf{a})} \;=\; \prod_{i=1}^{\ell} u_i^{\mathbf{v}_i \cdot \mathbf{a} + c_i} \;=\; \prod_{i=1}^{\ell} u_i^{c_i} \cdot \prod_{i=1}^{\ell} u_i^{\mathbf{v}_i \cdot \mathbf{a}} \;=\; u^c \cdot t^{\mathbf{a}} \quad .$$

Thus $\psi(u) = u^c \varphi_{\mathscr{A}}(f_\Delta(u))$, as maps to $\mathbb{R}^{\mathscr{A}}_>$. The common factor u^c does not affect the image in $\mathbb{R}^{\mathscr{A}}$, which shows that $\psi = \varphi_{\mathscr{A}} \circ f_\Delta$ and proves the theorem. □

A consequence of this proof of Theorem 3 is the derivation of equations which define the points of $X_{\mathscr{A}}$. This derivation is similar to, but easier than, the development of toric ideals in [19, Ch. 4], as we have monomials with arbitrary real-number exponents.

Suppose that we have a linear relation among the points of $\mathscr{A}$,

$$\sum_{\mathbf{a} \in \mathscr{A}} \mu_{\mathbf{a}} \cdot \mathbf{a} \;=\; \sum_{\mathbf{a} \in \mathscr{A}} \nu_{\mathbf{a}} \cdot \mathbf{a} \quad , \tag{16}$$

for some $\mu, \nu \in \mathbb{R}^{\mathscr{A}}$. Then the analytic binomial

$$\prod_{\mathbf{a} \in \mathscr{A}} z_{\mathbf{a}}^{\mu_{\mathbf{a}}} \;-\; \prod_{\mathbf{a} \in \mathscr{A}} z_{\mathbf{a}}^{\nu_{\mathbf{a}}} \;=:\; z^\mu \;-\; z^\nu \tag{17}$$

vanishes on $\varphi_{\mathscr{A}}(\mathbb{R}^d_>)$, considered as a point in $\mathbb{R}^{\mathscr{A}}_>$. This follows from the easy calculation

$$\varphi^*_{\mathscr{A}}(z^\mu) \;=\; \prod_{\mathbf{a}\in\mathscr{A}} (x^{\mathbf{a}})^{\mu_{\mathbf{a}}} \;=\; x^{\sum_{\mathbf{a}} \mu_{\mathbf{a}}\cdot\mathbf{a}} \;\;.$$

Even after clearing denominators, the common zero set of the binomials (17) is not exactly the image $\varphi_{\mathscr{A}}(\mathbb{R}^d_>)$ in the simplex $\Delta^{\mathscr{A}}$, as the point $\varphi_{\mathscr{A}}(x) \in \Delta^{\mathscr{A}}$ is where the ray

$$\mathbb{R}_> \cdot (x^{\mathbf{a}} \mid \mathbf{a} \in \mathscr{A}) \;\subset\; \mathbb{R}^{\mathscr{A}}_>$$

meets the simplex $\Delta^{\mathscr{A}}$. If we require that the binomial (17) is homogeneous in that $\sum_{\mathbf{a}\in\mathscr{A}} \mu_{\mathbf{a}} = \sum_{\mathbf{a}\in\mathscr{A}} \nu_{\mathbf{a}}$, then it vanishes at every point of this ray and therefore on the image of $\varphi_{\mathscr{A}}$ in $\Delta^{\mathscr{A}}$. Since the coordinates are positive numbers, we may further assume that (16) is an affine relation in that

$$\sum_{\mathbf{a}\in\mathscr{A}} \mu_{\mathbf{a}} \;=\; \sum_{\mathbf{a}\in\mathscr{A}} \nu_{\mathbf{a}} \;=\; 1 \;\;.$$

These necessary conditions are also sufficient.

Proposition 18. *A point z in $\Delta^{\mathscr{A}}$ lies in $X_{\mathscr{A}}$ if and only if we have*

$$z^\mu \;-\; z^\nu \;=\; 0$$

for all $\mu, \nu \in \mathbb{R}^{\mathscr{A}}$ with $\sum_{\mathbf{a}\in\mathscr{A}} \mu_{\mathbf{a}} = \sum_{\mathbf{a}\in\mathscr{A}} \nu_{\mathbf{a}} = 1$ and

$$\sum_{\mathbf{a}\in\mathscr{A}} \mu_{\mathbf{a}}\cdot\mathbf{a} \;=\; \sum_{\mathbf{a}\in\mathscr{A}} \nu_{\mathbf{a}}\cdot\mathbf{a} \;\;.$$

One way to see the sufficiency is to pick an affinely independent subset $\mathscr{C}$ of $\mathscr{A}$ that affinely spans $\mathbb{R}^d$ and use the formula $x^{\mathbf{a}} = z_{\mathbf{a}}$ for $\mathbf{a} \in \mathscr{C}$ to solve for $x \in \mathbb{R}^d$. Then the point $z \in \Delta^{\mathscr{A}}$ satisfies this collection of binomials if and only if $z = \varphi_{\mathscr{A}}(x)$. This is also evident if we take logarithms of the coordinates.

These arguments only work for points z in the interior of $\Delta^{\mathscr{A}}$. For points of $X_{\mathscr{A}}$ with some coordinates zero, we use the recursive nature of polytopes and the toric Bézier functions. Namely, if we restrict the collection $\{\beta_{\mathbf{a}} \mid \mathbf{a} \in \mathscr{A}\}$ of toric Bézier functions to a face $\mathscr{F}$ of the convex hull of $\mathscr{A}$, then those whose index $\mathbf{a}$ does not lie in $\mathscr{F}$ vanish, while those indexed by points of $\mathscr{A}$ lying in $\mathscr{F}$ specialize to toric Bézier functions for $\mathscr{F}$.

Theorem 16. *Let $\mathscr{A} \subset \mathbb{R}^d$, $w \in \mathbb{R}^{\mathscr{A}}_>$, and $\mathscr{B} \subset \mathbb{R}^n$ be exponents, weights, and control points for a toric Bézier patch. Suppose that $\mathscr{T}$ is a regular triangulation of $\mathscr{A}$ induced by a lifting function $\lambda : \mathscr{A} \to \mathbb{R}$. For each $t > 0$, let $F_t : \Delta \to \mathbb{R}^n$ be the toric Bézier patch of shape $\mathscr{A}$ with control points $\mathscr{B}$ and weights $t^{\lambda(\mathbf{a})} w_{\mathbf{a}}$. Then, for any $\varepsilon > 0$ there exists a t_0 such that if $t > t_0$, the image $F_t(\Delta)$ lies within ε of the control polytope $\mathscr{B}(\mathscr{T})$.*

Proof. The lifting function $\lambda : \mathscr{A} \to \mathbb{R}$ inducing the triangulation $\mathscr{T}$ also induces an action of $\mathbb{R}_>$ on $\Delta^{\mathscr{A}}$ where $t \in \mathbb{R}_>$ acts on a point $z \in \Delta^{\mathscr{A}}$ by scaling its coordinates, $(t.z)_{\mathbf{a}} = t^{\lambda(\mathbf{a})} z_{\mathbf{a}}$. Then $F_t(\Delta)$ is the image of $t.X_{\mathscr{A},w}$ under the projection $\pi_{\mathscr{B}} : \Delta^{\mathscr{A}} \to \mathbb{R}^n$. It suffices to show that $t.X_{\mathscr{A},w}$ can be made arbtrarily close to $|\mathscr{T}| \subset \Delta^{\mathscr{A}}$, if we choose t large enough.

We single out some equations from Proposition 18. Suppose that $\mathbf{a}, \mathbf{b} \in \mathscr{A}$ are points that do not lie in a common simplex of $\mathscr{T}$. That is, the segment $\overline{\mathbf{a},\mathbf{b}}$ is not an edge in the triangulation $\mathscr{T}$, and therefore it meets the interior of some face $\mathscr{F}$ of $\mathscr{T}$ so that

there is a point common to the interiors of $\mathscr{F}$ and of $\overline{\mathbf{a},\mathbf{b}}$. (If $\mathbf{a}=\mathbf{b}$, so that $\mathbf{a}$ does not participate in the triangulation $\mathscr{T}$, then this point is just $\mathbf{a}$.) This gives the equality of convex combinations

$$\mu\mathbf{a}+(1-\mu)\mathbf{b} \;=\; \sum_{\mathbf{c}\in\mathscr{F}} \nu_{\mathbf{c}}\cdot\mathbf{c} \;, \tag{18}$$

where all coefficients are positive and $\sum_{\mathbf{c}} \nu_{\mathbf{c}} = 1$. Thus

$$z_{\mathbf{a}}^{\mu} z_{\mathbf{b}}^{1-\mu} \;=\; \prod_{\mathbf{c}\in\mathscr{F}} z_{\mathbf{c}}^{\nu_{\mathbf{c}}}$$

holds on $X_{\mathscr{A}}$. The corresponding equation on $t.X_{\mathscr{A},w}$ is

$$z_{\mathbf{a}}^{\mu} z_{\mathbf{b}}^{1-\mu} \;=\; t^{\mu\mathbf{a}+(1-\mu)\mathbf{b}-\sum_{\mathbf{c}\in\mathscr{F}}\nu_{\mathbf{c}}\cdot\mathbf{c}} \cdot \frac{w_{\mathbf{a}}^{\mu} w_{\mathbf{b}}^{1-\mu}}{\prod_{\mathbf{c}\in\mathscr{F}} w_{\mathbf{c}}^{\nu_{\mathbf{c}}}} \cdot \prod_{\mathbf{c}\in\mathscr{F}} z_{\mathbf{c}}^{\nu_{\mathbf{c}}} \;. \tag{19}$$

Since $\overline{\mathbf{a},\mathbf{b}}$ is not in the triangulation, points in the interior of the lifted segment

$$\overline{(\mathbf{a},\lambda(\mathbf{a})),\,(\mathbf{b},\lambda(\mathbf{b}))}$$

lie below points of upper faces of the polytope P_λ. We apply this observation to the point (18). Its height in the lifted segment is $\mu\mathbf{a}+(1-\mu)\mathbf{b}$, while its height in the lift of the face $\mathscr{F}$ is $\sum_{\mathbf{c}\in\mathscr{F}} \nu_{\mathbf{c}}\cdot\mathbf{c}$, and so

$$\mu\mathbf{a}+(1-\mu)\mathbf{b} \;<\; \sum_{\mathbf{c}\in\mathscr{F}} \nu_{\mathbf{c}}\cdot\mathbf{c} \;.$$

This implies that the exponent of t in (19) is negative. Since the other terms on the right hand side are bounded, we see that the left hand side, and in fact the simple product $z_{\mathbf{a}}z_{\mathbf{b}}$, may be made as small as we please by requiring that t be sufficiently large.

Suppose that $\mathscr{A}$ consists of $\ell+d+1$ elements. Repeating the previous argument for the (finitely many) pairs of points $\mathbf{a},\mathbf{b}$ which are not both in any simplex of $\mathscr{T}$, we see that for any $\varepsilon>0$, there is a t_0 such that if $t>t_0$ and $z\in X_{\mathscr{A},w}$, then

$$z_{\mathbf{a}}z_{\mathbf{b}} \;<\; \varepsilon^2/4\ell^2 \;,$$

whenever $\mathbf{a},\mathbf{b}$ do not lie in a common simplex of $\mathscr{T}$. In particular, at most one of $z_{\mathbf{a}}$ or $z_{\mathbf{b}}$ can exceed $\varepsilon/2\ell$.

Let $z\in X_{\mathscr{A},w}$. Then there is some facet $\mathscr{F}$ of $\mathscr{T}$ such that if $\mathbf{a}\notin\mathscr{F}$, then $0\le z_{\mathbf{a}} < \varepsilon/2\ell$. Suppose that $\mathscr{F}=\{\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_d\}$ and set

$$z_{\mathscr{F}} \;:=\; (1-z_{\mathbf{a}_1}-\cdots-z_{\mathbf{a}_d})e_{\mathbf{a}_0}+z_{\mathbf{a}_1}e_{\mathbf{a}_1}+\cdots+z_{\mathbf{a}_d}e_{\mathbf{a}_d} \;,$$

which is a point of the facet $|\mathscr{F}|$ of the geometric realization $|\mathscr{T}|\subset\Delta^{\mathscr{A}}$. Then

$$\begin{aligned}
\|z-z_{\mathscr{F}}\| &= \left\| \sum_{\mathbf{a}\in\mathscr{A}} z_{\mathbf{a}}e_{\mathbf{a}} \;-\; (1-z_{\mathbf{a}_1}-\cdots-z_{\mathbf{a}_d})e_{\mathbf{a}_0} - z_{\mathbf{a}_1}e_{\mathbf{a}_1} - \cdots - z_{\mathbf{a}_d}e_{\mathbf{a}_d} \right\| \\
&\le \sum_{\mathbf{a}\notin\mathscr{F}} z_{\mathbf{a}} \;+\; (1-z_{\mathbf{a}_0}-\cdots-z_{\mathbf{a}_d}) \;=\; 2\sum_{\mathbf{a}\notin\mathscr{F}} z_{\mathbf{a}} \\
&< 2\ell\frac{\varepsilon}{2\ell} \;=\; \varepsilon \,,
\end{aligned}$$

as $1=\sum_{\mathbf{a}} z_{\mathbf{a}}$. □

We say that two subsets X and Y of Euclidean space are within a distance ε if for every point x of X there is some point y of Y within a distance ε of x, and vice-versa.

Theorem 17. *Let $\mathscr{A} \subset \mathbb{R}^d$ be a finite set of exponents. Suppose that $|\mathscr{T}| \subset \triangle^{\mathscr{A}}$ is the geometric realization of a triangulation $\mathscr{T}$ of $\mathscr{A}$ and there is a weight w such that the distance between $X_{\mathscr{A},w}$ and $|\mathscr{T}|$ is less than $1/2(d+1)$. Then $\mathscr{T}$ is the regular triangulation induced by the lifting function $\lambda(\mathbf{a}) = \log(w_{\mathbf{a}})$.*

Proof. To show that $\mathscr{T}$ is the regular triangulation induced by the lifting function λ whose value at $\mathbf{a}$ is $\log(w_{\mathbf{a}})$, we must show that if a segment $\overline{\mathbf{a},\mathbf{b}}$ between two points of $\mathscr{A}$ does not lie in the triangulation $\mathscr{T}$, then its lift by λ lies below the lift of some face $\mathscr{F}$ of $\mathscr{T}$.

Set $\varepsilon := 1/2(d+1)$. For each face $\mathscr{F}$ of $\mathscr{T}$, let $x_{\mathscr{F}} \in \triangle^{\mathscr{A}}$ be the barycenter of $\mathscr{F}$,

$$x_{\mathscr{F}} \;:=\; \sum_{\mathbf{a}\in\mathscr{F}} \frac{1}{\#\mathscr{F}} e_{\mathbf{a}} \;\;,$$

where $\#\mathscr{F}$ is the number of points of $\mathscr{A}$ in $\mathscr{F}$, which is at most $d+1$. If z is a point of $X_{\mathscr{A},w}$ within a distance ε of $x_{\mathscr{F}}$, so that $\|z - x_{\mathscr{F}}\| < \varepsilon$, then in particular no component of the vector $z - x_{\mathscr{F}}$ has absolute value exceeding ε. Thus we have the dichotomy

$$\begin{array}{ll} z_{\mathbf{a}} < & \varepsilon \;=\; 1/2(d+1) \qquad \text{if } \mathbf{a} \notin \mathscr{F}\,, \\ z_{\mathbf{a}} > 1/\#\mathscr{F} - \varepsilon > 1/2(d+1) & \text{if } \mathbf{a} \in \mathscr{F} \;\;. \end{array} \tag{20}$$

Now suppose that the segment $\overline{\mathbf{a},\mathbf{b}}$ does not lie in the triangulation $\mathscr{T}$. Then there is a face $\mathscr{F}$ of the triangulation whose interior meets the interior of this segment. That is, there is an equality of convex combinations (18) and a corresponding equation that holds for points $z \in X_{\mathscr{A},w}$,

$$z_{\mathbf{a}}^{\mu} z_{\mathbf{b}}^{1-\mu} \cdot \prod_{\mathbf{c}\in\mathscr{F}} w_{\mathbf{c}}^{\nu_{\mathbf{c}}} \;=\; w_{\mathbf{a}}^{\mu} w_{\mathbf{b}}^{1-\mu} \cdot \prod_{\mathbf{c}\in\mathscr{F}} z_{\mathbf{c}}^{\nu_{\mathbf{c}}} \;\;.$$

Suppose that z is a point of $X_{\mathscr{A},w}$ that lies within a distance of $1/2(d+1)$ of the barycenter $X_{\mathscr{F}}$ of the face $\mathscr{F}$. Then, by the estimates (20), we have

$$\frac{1}{2(d+1)} \prod_{\mathbf{c}\in\mathscr{F}} w_{\mathbf{c}}^{\nu_{\mathbf{c}}} \;>\; z_{\mathbf{a}}^{\mu} z_{\mathbf{b}}^{1-\mu} \cdot \prod_{\mathbf{c}\in\mathscr{F}} w_{\mathbf{c}}^{\nu_{\mathbf{c}}} \;=\; w_{\mathbf{a}}^{\mu} w_{\mathbf{b}}^{1-\mu} \cdot \prod_{\mathbf{c}\in\mathscr{F}} z_{\mathbf{c}}^{\nu_{\mathbf{c}}} \;>\; w_{\mathbf{a}}^{\mu} w_{\mathbf{b}}^{1-\mu} \cdot \frac{1}{2(d+1)} \;\;.$$

canceling the common factor of $1/2(d+1)$ and taking logarithms, we obtain

$$\sum_{\mathbf{c}\in\mathscr{F}} \nu_{\mathbf{c}} \log(w_{\mathbf{c}}) \;>\; \mu \log(w_{\mathbf{a}}) + (1-\mu)\log(w_{\mathbf{b}}) \;\;,$$

which implies that the point (18) common to the segment $\overline{\mathbf{a},\mathbf{b}}$ and the face $\mathscr{F}$ of $\mathscr{T}$ is lifted higher in the face $\mathscr{F}$ than in the segment $\overline{\mathbf{a},\mathbf{b}}$, and so the lift of the segment $\overline{\mathbf{a},\mathbf{b}}$ by λ lies below the lift of the face $\mathscr{F}$. As this is true for all segments, we see that $\mathscr{T}$ is the triangulation induced by the lifting function λ. □

A Comparison of Different Progressive Iteration Approximation Methods

Jorge Delgado[1] and Juan Manuel Peña[2]

[1] Dept. Applied Mathematics, University of Zaragoza, Escuela Universitaria Politécnica de Teruel, 44003 - Teruel, Spain
jorgedel@unizar.es
[2] Dept. Applied Mathematics/IUMA, University of Zaragoza, 50009 - Zaragoza, Spain

Abstract. The progressive iteration approximation property (PIA) permits applying tools of CAGD to the interpolation of curves and surfaces. Convergence rates of PIA for bases of different spaces are analyzed and compared, including tables of rate of convergence and figures. In the rational case, the importance of an adequate choice of the weights and of an adequate evaluation algorithm is shown. Finally, a new approach to the PIA is introduced.

1 Introduction

In many applications of Computer Aided Geometric Design (CAGD) it is necessary to interpolate or approximate a data point set with a parametric curve or a parametric surface. In the literature several methods have been considered for this purpose. One of them uses the progressive iteration approximation property of a curve and a surface (see [5], [7], [10], [9] and [13]). This paper deals with this approach. In [10] the application of the progressive iteration approximation method to normalized totally positive representations was analyzed. In [7] the convergence rates of the different normalized totally positive bases of a space of functions were compared, and the bases with the highest convergence rate were obtained. Here we compare the convergence rate of the optimal bases of different spaces of functions.

Let us recall that a basis $(u_0,\dots,u_n)$ defined on a set I formed by nonnegative functions and such that $\sum_{i=0}^n u_i(x)=1$ for all $x\in I$ is called a *blending basis*. Given a sequence of points $\{P_i\}_{i=0}^m$ whose i^{th} point is assigned to a parameter value t_i, for $i=0,1,\dots,m$, and a blending basis $(u_0,\dots,u_m)$, we generate a starting curve as $\gamma^0(t)=\sum_{i=0}^m P_i^0 u_i(t)$ where $P_i^0=P_i$ for $i=0,1,\dots,m$. Then, by calculating the adjusting vector for each control point $\Delta_i^0=P_i-\gamma^0(t_i)$, and taking $P_i^1=P_i^0+\Delta_i^0$ for $i=0,1,\dots,m$, we get the curve $\gamma^1(t)=\sum_{i=0}^m P_i^1 u_i(t)$. Iterating this process we get a sequence of curves $\{\gamma^k\}_{k=0}^\infty$. Then, when $\lim_{k\to\infty}\gamma^k(t_i)=P_i$ for all $i\in\{0,1,\dots,m\}$ we say that the initial curve satisfies the *progressive iteration approximation* (PIA) property.

Given a sequence of points $\{P_{ij}\}_{0\le i\le m}^{0\le j\le n}$, such that the point P_{ij} is assigned to a couple of parameter values (x_i,y_j), for $i=0,1,\dots,m$ and $j=0,1,\dots,n$, and a blending basis $(u_{ij})_{0\le i\le m}^{0\le j\le n}$, we generate a starting surface as $F^0(x,y)=\sum_{i=0}^m\sum_{j=0}^n P_{ij}^0 u_{ij}(x,y)$ where $P_{ij}^0=P_{ij}$ for $i=0,1,\dots,m$ and $j=0,1,\dots,n$. Then, by calculating the adjusting vector for each control point $\Delta_{ij}^0=P_{ij}-F^0(x_i,y_j)$, and taking $P_{ij}^1=P_{ij}^0+\Delta_{ij}^0$ for $i=0,1,\dots,m$

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 136–152, 2010.

and $j = 0,1,\ldots,n$, we get the curve $F^1(x,y) = \sum_{i=0}^{m}\sum_{j=0}^{n} P_{ij}^1 u_{ij}(x,y)$. Iterating this process we get a sequence of surfaces $\{F^k\}_{k=0}^{\infty}$. When $\lim_{k\to\infty} F^k(x_i,y_j) = P_{ij}$ for all $i \in \{0,1,\ldots,m\}$ and $j \in \{0,1,\ldots,n\}$, we say that the initial surface satisfies the *progressive iteration approximation* (PIA) property.

Section 2 recalls the main results on the PIA property. Section 3 analyzes the convergence rates with the Bernstein basis and motivates the use of B-splines. It also shows graphically and numerically the high convergence rate of the PIA when using B-splines. Section 4 considers rational bases and shows, for low degrees, the optimal choice of the weights in order to get the fastest convergence rate of PIA property. Section 5 presents a new approach of the PIA. Conclusions and future work are discussed in Section 6.

2 Background of the Progressive Iteration Approximation Property

In [10] it was studied in a general way under what conditions the curves generated through a blending system $(u_0,\ldots,u_n)$ satisfy the PIA property.

Before stating this general result we need to define some concepts. The *collocation matrix* of a system of functions $(u_0(t),\ldots,u_n(t))$ at a sequence of ordered points $t_0 < \cdots < t_n$ in $\mathbb{R}$ is given by

$$M\begin{pmatrix} u_0,\ldots,u_n \\ t_0,\ldots,t_m \end{pmatrix} := (u_j(t_i))_{i=0,\ldots,m;j=0,\ldots,n}. \tag{1}$$

A matrix is *totally positive (TP)* if all its minors are nonnegative and a system of functions is TP when all its collocation matrices (1) are TP. If, in addition, it is a blending basis, then it is called normalized totally positive (NTP). In case of an NTP basis one knows that the curve imitates the shape of its control polygon, due to the variation diminishing properties of TP matrices (see [11]). In fact, shape preserving representations are associated with NTP bases.

An algorithm for the PIA procedure for curves can be written in the following way:

Algorithm 1. *PIA algorithm for curves.*

Input: $(P_i)_{i=0}^n$, $(t_i)_{i=0}^n$, $(u_0,\ldots,u_n)$, *nTol, xTol*

For $i = 0:n$
$\quad P_i^0 = P_i$
End-For

$\gamma^0(t) = \sum_{i=0}^{n} P_i^0 u_i(t)$

$k = 1$

$\gamma^{-1}(t_i) = \infty \; i = 0,1,\ldots,n$

While $k \le nTol$ *and* $\max_{0\le i\le n} \|P_i - \gamma^{k-1}(t_i)\| \ge xTol$
$\quad$ *For* $i = 0:n$
$\quad\quad \Delta_i^{k-1} = P_i - \gamma^{k-1}(t_i)$
$\quad\quad P_i^k = P_i^{k-1} + \Delta_i^{k-1}$

End-For

$\gamma^k(t) = \sum_{i=0}^n P_i^k u_i(t)$

$k = k+1$

End-While

Output: $\gamma^{k-1}(t)$ *and, if nTol is exceeded, a warning message.*

Remark 1. Let us observe that in Algorithm 1 it is necessary to evaluate curves of the form $\gamma^k(t) = \sum_{i=0}^n P_i^k u_i(t)$. The algorithm used for this depends on the system $(u_0,\dots,u_n)$. For example, if $(u_0,\dots,u_n)$ is the Bernstein basis of the space of polynomials of degree less than or equal to n the corresponding curves are evaluated through the de Casteljau algorithm.

The following result provides sufficient conditions in order to ensure the convergence of Algorithm 1.

Theorem 2. *(cf. Theorem 2.1 of [10]) Let* $U = (u_0,\dots,u_n)$ *be a blending basis. Then, if the basis* U *is TP and its collocation matrix* M *at* $t_0 < t_1 < \cdots < t_n$ *is nonsingular, the curve* $\gamma^0(t) = \sum_{i=0}^n P_i^0 u_i(t)$ *satisfies the progressive iteration approximation property.*

In [10] it was also analyzed in a general way under what conditions the surfaces generated through a tensor product basis satisfies the PIA property.

Iterating the PIA procedure for surfaces described in Section 1, an algorithm in pseudo-code can be written in the following way:

Algorithm 2. *PIA algorithm for surfaces.*

Input: $(P_{ij})_{\substack{0\le j\le n\\ 0\le i\le m}}$, $(x_i)_{i=0}^m$, $(y_j)_{j=0}^n$, $(u_0,\dots,u_m)$, $(\overline{u}_0,\dots,\overline{u}_n)$, *nTol, xTol*

For $i = 0:m$

For $j = 0:n$

$P_{ij}^0 = P_{ij}$

End-For

End-For

$F^0(x,y) = \sum_{i=0}^m \sum_{j=0}^n P_{ij}^0 u_i(x)\,\overline{u}_j(y)$

$k = 1$

$F^{-1}(x_i,y_j) = \infty\ i = 0,1,\dots,m$ *and* $j = 0,1,\dots,n$

While $k \le nTol$ *and* $\max_{0\le i\le m;0\le j\le n} \|P_{ij} - F^{k-1}(x_i,y_j)\| \ge xTol$

For $i = 0:m$

For $j = 0:n$

$\Delta_{ij}^{k-1} = P_{ij} - F^{k-1}(x_i,y_j)$

$P_{ij}^k = P_{ij}^{k-1} + \Delta_{ij}^{k-1}$

End-For

End-For

$F^k(x,y) = \sum_{i=0}^m \sum_{j=0}^n P_{ij}^k u_i(x)\,\overline{u}_j(y)$

$k = k+1$
End-While

Output: $F^{k-1}(x,y)$ *and, if nTol is exceeded, a warning message.*

The following result provides sufficient conditions in order to ensure the convergence of Algorithm 2.

Theorem 3. *(cf. Theorem 2.2 of [10]) Let* $U = (u_0,\dots,u_m)$ *and* $\overline{U} = (\overline{u}_0,\dots,\overline{u}_n)$ *be two blending bases. Then, if the bases* U *and* $\overline{U}$ *are TP and the collocation matrices of* U *at* $x_0 < x_1 < \cdots < x_m$ *and of* $\overline{U}$ *at* $y_0 < y_1 < \cdots < y_n$ *are nonsingular, the tensor product surface* $F^0(x,y) = \sum_{i=0}^{m}\sum_{j=0}^{n} P_{ij}^0 u_i(x)\overline{u}_j(y)$ *satisfies the progressive iteration approximation property.*

In order to find the bases with fastest convergence rates in a given space with NTP bases we need to introduce the following concept.

Let $(u_0,\dots,u_n)$ be a TP basis of a vector space of functions $\mathscr{U}$. Then we say that $(u_0,\dots,u_n)$ is a *B-basis* if for any other totally positive basis $(v_0,\dots,v_n)$ of $\mathscr{U}$ the matrix K of change of basis

$$(v_0,\dots,v_n) = (u_0,\dots,u_n)K$$

is TP. Let us recall that, if a vector space of functions has a TP (resp., NTP) basis then it has a B-basis (resp., normalized B-basis). It has been proved that in a vector space of functions with NTP bases its normalized B-basis has optimal shape preserving properties among all the NTP bases of the space (see [3]).

Using some strong results of Linear Algebra, Theorem 4 of [7] shows the basis with the fastest convergence rate among all NTP bases of a space of functions.

Theorem 4. *Given a space of functions* $\mathscr{U}$ *with an NTP basis, the normalized B-basis provides a PIA with the fastest convergence rate among all NTP bases of* $\mathscr{U}$.

Now we state the analogous result for tensor product surfaces.

Theorem 5. *(cf. Corollary 6 of [7]) Given spaces of functions* $\mathscr{U}$ *and* $\mathscr{V}$ *with NTP bases, the tensor product of the normalized B-bases of* $\mathscr{U}$ *and* $\mathscr{V}$ *provides a progressive iterative approximation with the fastest convergence rate among all bases which are tensor product of NTP bases of* $\mathscr{U}$ *and* $\mathscr{V}$.

From now on we will compare the convergence rates of B-bases of different spaces of functions.

3 PIA Algorithms Using the Bernstein and the Uniform B-Spline Bases

One of the most important bases in CAGD is the Bernstein basis. It is well-known that the Bernstein basis of degree n defined on $[0,1]$ is the normalized B-basis of the space of polynomials of degree less than or equal to n (see [2]). Therefore, by Theorem 4, the

Bernstein basis of degree n on $[0,1]$ provides a PIA with the fastest convergence rates among the NTP bases of the space of polynomials of degree less than or equal to n. The Bernstein basis $(b_0^n,\dots,b_n^n)$ of the space of polynomials of degree at most n is defined by

$$b_i^n(t)=\binom{n}{i}t^i(1-t)^{n-i},\quad t\in[0,1],$$

for $i=0,1,\dots,n$.

In order to prove Theorem 2, Algorithm 1 was written in matrix form. Since this matrix form is also very useful to study the convergence rate of the method, let us recall it:

$$(\Delta_0^k,\Delta_1^k,\dots,\Delta_n^k)^T=(I-M)^k(\Delta_0^0,\Delta_1^0,\dots,\Delta_n^0). \tag{2}$$

where I is the identity matrix of $n+1$ order,

$$M:=M\begin{pmatrix}u_0,\dots,u_n\\ t_0,\dots,t_n\end{pmatrix}$$

and the Δ_j^l's are row-vectors. Therefore, the iterative method converges if and only if $\lim_{k\to\infty}\Delta_i^k=0$ for all $i\in\{0,1,\dots,n\}$. Then, from (2) we deduce that the PIA converges if and only if $\rho(I-M)<1$, where $\rho(B)$ denotes the spectral radius of B. All the eigenvalues of a nonsingular totally positive matrix are real and belong to the interval $(0,1]$ (see Corollary 6.6 of [1]). If $\lambda_0,\lambda_1,\dots,\lambda_n$ are the $n+1$ eigenvalues of M then, the $n+1$ eigenvalues $\widetilde{\lambda}_0,\widetilde{\lambda}_1,\dots,\widetilde{\lambda}_n$ of $I-M$ are given by $\widetilde{\lambda}_i=1-\lambda_i$, for $i=0,1,\dots,n$. Hence, we deduce that $\rho(I-M)=1-\min_{0\le i\le n}\lambda_i$. Obviously, the higher $\min_{0\le i\le n}\lambda_i$ is, the faster Algorithm 1 is.

First let us consider the application of Algorithm 1 with $(u_0,\dots,u_n)=(b_0^n,\dots,b_n^n)$ and the sequence $(t_i)_{i=0}^n$ given by $t_i=i/n$ for $i=0,1,\dots,n$. As we have just seen, the convergence rate of the method depends on the minimal eigenvalue of the corresponding collocation matrix.

Theorem 6. *The $n+1$ eigenvalues $\lambda_0,\lambda_1,\dots,\lambda_n$ of the matrix*

$$M=M\begin{pmatrix}b_0^n(t),\dots,b_n^n(t)\\ t_0,\dots,t_n\end{pmatrix}, \tag{3}$$

with $t_i=i/n$ for $i=0,1,\dots,n$, are given by

$$\lambda_i=\frac{n!}{(n-i)!}\frac{1}{n^i},\quad i=0,1,\dots,n. \tag{4}$$

Proof. The Bernstein operator is given by

$$\mathscr{B}^n[g]=\sum_{i=0}^n g(t_i)\,b_i^n(t)=\sum_{i=0}^n g(i/n)\,b_i^n(t).$$

Let λ be an eigenvalue of the matrix M and let $v=(v_0,v_1,\dots,v_n)$ be its associated eigenvector. Now let us prove that λ is also an eigenvalue of the Bernstein operator associated to the eigenfunction $f(t)=\sum_{i=0}^n v_i b_i^n(t)$. Since λ is an eigenvalue of the

matrix M and $v = (v_0, v_1, \ldots, v_n)$ its associated eigenvector, we have that $M\,v = \lambda\,v$, and so from formula (3) we deduce that

$$\begin{pmatrix} \sum_{i=0}^{n} v_i\, b_i^n(t_0) \\ \sum_{i=0}^{n} v_i\, b_i^n(t_1) \\ \vdots \\ \sum_{i=0}^{n} v_i\, b_i^n(t_n) \end{pmatrix} = \begin{pmatrix} \lambda\, v_0 \\ \lambda\, v_1 \\ \vdots \\ \lambda\, v_n \end{pmatrix}.$$

Then, multiplying the previous expression by the Bernstein basis, gathered as a row matrix $(b_0^n(t), b_1^n(t), \ldots, b_n^n(t))$, we deduce that

$$\sum_{i=0}^{n} f(t_i)\, b_i^n(t) = \lambda\, f(t).$$

Hence λ is an eigenvalue of the Bernstein operator and f is its associated eigenfunction.

Conversely, let $\widetilde{\lambda}$ be an eigenvalue of the Bernstein operator and let $\widetilde{f}$ be its associated eigenfunction. Since $\widetilde{\lambda}$ is an eigenvalue of the Bernstein operator $\mathscr{B}^n$ and $\widetilde{f}$ its associated eigenfunction, we have that

$$\sum_{i=0}^{n} \widetilde{f}(t_i)\, b_i^n(t) = \widetilde{\lambda}\, \widetilde{f}(t).$$

Then, evaluating the previous expression at $t_0, t_1, \ldots, t_n$, we have that

$$\sum_{i=0}^{n} \widetilde{f}(t_i)\, b_i^n(t_k) = \widetilde{\lambda}\, \widetilde{f}(t_k), \quad \text{for } k = 0, 1, \ldots, n.$$

We can write the previous formulas in matrix form as

$$M \begin{pmatrix} b_0^n(t), \ldots, b_n^n(t) \\ t_0, \ldots, t_n \end{pmatrix} \begin{pmatrix} \widetilde{f}(t_0) \\ \widetilde{f}(t_1) \\ \vdots \\ \widetilde{f}(t_n) \end{pmatrix} = \widetilde{\lambda} \begin{pmatrix} \widetilde{f}(t_0) \\ \widetilde{f}(t_1) \\ \vdots \\ \widetilde{f}(t_n) \end{pmatrix}.$$

Therefore, $\widetilde{\lambda}$ is also an eigenvalue of the matrix M and $v = (\widetilde{f}(t_0), \widetilde{f}(t_1), \ldots, \widetilde{f}(t_n))$ is its associated eigenvector.

It is well known that the eigenvalues of this operator are given by formula (4) (see [4]). Therefore, we can conclude that the eigenvalues $\lambda_0, \lambda_1, \ldots, \lambda_n$ of the collocation matrix in (3) are also given by formula (4).

Let us observe that the previous result does not hold if $t_0 < \cdots < t_n$ are arbitrary points in $[0,1]$. In fact, if $n = 1$, $t_0 = 0.9$ and $t_1 = 1$, then the matrix

$$M \begin{pmatrix} b_0^1(t), b_1^1(t) \\ t_0, t_1 \end{pmatrix} = \begin{pmatrix} 0.1 & 0.9 \\ 0 & 1 \end{pmatrix}$$

has eigenvalues $0.1, 1$ instead of $1, 1$.

In order to prove Theorem 3, Algorithm 2 was written in matrix form. For this purpose, it is necessary to recall the Kronecker product of two matrices: given two matrices $A=(a_{ij})_{1\le i\le m}^{0\le j\le n}$ and $B=(b_{ij})_{1\le i\le p}^{0\le j\le q}$, their Kronecker product, denoted by $A\otimes B$, is defined to be the block matrix

$$A\otimes B=\begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix},$$

and so $A\otimes B$ is an $(mp)\times(nq)$ matrix (see [8] for more details). So, we can write Algorithm 2 in matrix form as

$$\begin{aligned}&(\Delta_{00}^{k+1},\Delta_{01}^{k+1},\ldots,\Delta_{0n}^{k+1},\Delta_{10}^{k+1},\ldots,\Delta_{1n}^{k+1},\ldots,\Delta_{m1}^{k+1},\ldots,\Delta_{mn}^{k+1})^T\\ &\quad=(I-B)\,(\Delta_{00}^{k},\Delta_{01}^{k},\ldots,\Delta_{0n}^{k},\Delta_{10}^{k},\ldots,\Delta_{1n}^{k},\ldots,\Delta_{m1}^{k},\ldots,\Delta_{mn}^{k})^T,\end{aligned}$$

where I is the identity matrix of $(m+1)(n+1)$ order and the matrix B is the Kronecker product of the collocation matrices

$$B_1=M\begin{pmatrix} u_0,u_1,\ldots,u_m \\ x_0,x_1,\ldots,x_m \end{pmatrix}$$

and

$$B_2=M\begin{pmatrix} \overline{u}_0,\overline{u}_1,\ldots,\overline{u}_n \\ y_0,y_1,\ldots,y_n \end{pmatrix},$$

that is, $B=B_1\otimes B_2$. Therefore, the PIA property for surfaces converges if and only if $\rho(I-B)<1$. If the eigenvalues of a matrix B_1 are $\lambda_1,\ldots,\lambda_m$, and the eigenvalues of a matrix B_2 are $\mu_1,\ldots,\mu_n$, it is well-known that every eigenvalue of the Kronecker product B can be expressed as the product of an eigenvalue of B_1 and an eigenvalue of B_2, $\lambda_i\mu_j$ (see Theorem 4.2.12 of [8]). Therefore, $\rho(I-B)=1-(\min_{0\le i\le m}\lambda_i)(\min_{0\le j\le n}\mu_j)$. Obviously, the higher $\min_{0\le i\le m}\lambda_i$ and $\min_{0\le j\le n}\mu_j$ are, the faster Algorithm 2 is.

As we have seen, the PIA rate depends on the minimal eigenvalue of the collocation matrix of the corresponding basis. Obviously, the collocation matrix depends on the basis. So in [7] it was pointed out the fastest basis of a space of functions with a NTP basis.

We now apply Algorithm 2 taking $(u_0,\ldots,u_m)=(b_0^m,\ldots,b_m^m)$, $(\overline{u}_0,\ldots,\overline{u}_n)=(b_0^n,\ldots,b_n^n)$ and the sequences $(x_i)_{i=0}^m$ and $(y_j)_{j=0}^n$ given by $x_i=i/m$ for $i=0,1,\ldots,m$ and $y_j=j/n$ for $j=0,1,\ldots,n$. As we have seen previously, the convergence rate of the method depends on the eigenvalues of the corresponding collocation matrix.

Theorem 7. *The $(m+1)(n+1)$ eigenvalues $(\rho_{ij})_{0\le i\le m}^{0\le j\le n}$ of the matrix $M=U\otimes\overline{U}$ where*

$$U=M\begin{pmatrix} b_0^m(t),\ldots,b_m^m(t) \\ x_0,\ldots,x_m \end{pmatrix} \tag{5}$$

with $x_i=i/m$ for $i=0,1,\ldots,m$ and

$$\overline{U}=M\begin{pmatrix} b_0^n(t),\ldots,b_n^n(t) \\ y_0,\ldots,y_n \end{pmatrix} \tag{6}$$

with $y_j = j/n$ *for* $j = 0,1,\ldots,n$, *are given by* $\rho_{ij} = \lambda_i \overline{\lambda}_j$ *for* $i = 0,1,\ldots,m$ *and* $j = 0,1,\ldots,n$ *where*

$$\lambda_i = \frac{m!}{(m-i)!}\frac{1}{m^i}, \quad i = 0,1,\ldots,m$$

and

$$\overline{\lambda}_j = \frac{n!}{(n-i)!}\frac{1}{n^i}, \quad i = 0,1,\ldots,n.$$

Proof. The result is a straightforward consequence of Theorem 4 taking into account the fact that the eigenvalues of matrix B, which is the Kronecker product of two matrices B_1 and B_2, are products of an eigenvalue of B_1 by an eigenvalue of B_2.

In Table 1 we can see the minimal eigenvalue of the collocation matrix of the Bernstein basis of degree n (dimension $n+1$), $(b_0^n,\ldots,b_n^n)$, at the strictly increasing sequence of points $(t_i)_{0\le i\le n}$ given by $t_i = i/n$ for $i = 0,1,\ldots,n$ for different values of n.

Table 1. Minimal eigenvalue of the collocation matrix of the Bernstein basis of degree n

Degree n	Dimension	$\min_{0\le i\le n}\lambda_i$
1	2	$1.00000E+00$
2	3	$5.00000E-01$
3	4	$2.22222E-01$
4	5	$9.37500E-02$
5	6	$3.84000E-02$
10	11	$3.62880E-04$
15	16	$2.98628E-06$
20	21	$2.32019E-08$

Let us recall that the minimal eigenvalue of the corresponding collocation matrix determines the convergence rate of the PIA. As the degree of the Bernstein basis increases, the minimal eigenvalue decreases very quickly for both algorithms 1 (see Table 1) and 2. Hence, this method with the Bernstein basis is not interesting from a practical point of view because of the slow convergence. Let us show how the convergence rate of PIA increases considering the B-bases of another family of space of functions: the uniform B-spline bases (see [3]).

We first give the general definition of the B-spline basis. Let $U = \{u_0,u_1,\ldots,u_m\}$ be a nondecreasing sequence of real numbers: $u_0 \le u_1 \le \cdots \le u_m$. The numbers u_i are called *knots*, and U is the *knot vector*. The ith B-spline basis function of degree p (order $p+1$) over the knot vector U, denoted by $N_i^p(u)$, is defined recursively as

$$N_i^0(u) = \begin{cases} 1, \text{ if } u_i \le u < u_{i+1}, \\ 0, \text{ otherwise,} \end{cases}$$

and

$$N_i^p(u) = \frac{u-u_i}{u_{i+p}-u_i}N_i^{p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p-1}-u_{i+1}}N_{i+1}^{p-1}(u).$$

Remark 8. The previous equation can yield the quotient $0/0$. In this case we define the quotient as 0.

For more details on B-splines see [12].

In particular let us consider only the uniform B-spline bases of degree p over nonperiodic (or clamped or open) knot vectors $U = \{u_0, u_1, \ldots, u_m\}$, that is,

$$\begin{array}{ll} u_i = a, & \text{for } i = 0, 1, \ldots, p, \\ u_i = b, & \text{for } i = m-p, \ldots, m, \\ u_i = a + (i-p)\frac{b-a}{m-2p}, & \text{for } i = p+1, \ldots, m-p-1. \end{array} \tag{7}$$

The reason to consider only this sort of knot vectors is that the corresponding B-spline curves satisfy the endpoint interpolation property and then, if we take in Algorithm 1 the first term of the sequence $(t_i)_{i=0}^n$ as a and the last one as b, at least two of the eigenvalues are 1. In contrast to the case of polynomial curves, when applying PIA with B-splines the convergence rate keeps high even for high dimensional spaces if we keep a low degree.

First let us consider uniform B-spline bases of degree 2 over nonperiodic knot vectors $U = \{u_0, u_1, \ldots, u_m\}$ defined by (7) with $a = 0$ and $b = 1$. The dimension of the corresponding space of functions is $m-2$. So we will denote the B-spline basis obtained in this way by $(N_0^2, \ldots, N_{m-3}^2)$. Now, let us consider the strictly increasing sequence $(t_i)_{i=0}^{m-3}$ given by $t_i = i/(m-3)$. Then, we can apply Algorithm 1 with the basis and sequence previously considered. As we have recalled above, the minimal eigenvalue of the collocation matrix

$$M\begin{pmatrix} N_0^2, \ldots, N_{m-3}^2 \\ t_0, \ldots, t_{m-3} \end{pmatrix} \tag{8}$$

determines the convergence rate of this particular algorithm. In Table 2 we can see the minimal eigenvalue of the collocation matrices (8) for differents values of m.

Table 2. Minimal eigenvalue of the collocation matrix of the B-Spline basis of degree 2

m	Dimension	$\min_{0\le i\le n}\lambda_i$
12	10	$4.52981E-01$
22	20	$4.75086E-01$
32	30	$4.83271E-01$
42	40	$4.87435E-01$
52	50	$4.89946E-01$
62	60	$4.91623E-01$
72	70	$4.92822E-01$
82	80	$4.93721E-01$
92	90	$4.94420E-01$
102	100	$4.94979E-01$

Let us observe that the minimal eigenvalue of the matrices of the previous kind keep almost constant in spite of the dimension, which is very important from a practical point of view. In order to check the rate of convergence of this particular algorithm (see Table 2) we have applied it to two sets of points.

As an example, let us consider the Lemniscate of Gerono given by the parametric equations

$$\begin{cases} x(t) = \cos t, \\ y(t) = \sin t \cdot \cos t, \end{cases} \tag{9}$$

$t \in [0, 2\pi]$. We took a sample of 11 points $\{P_i\}_{i=0}^{10}$ from the Lemniscate of Gerono in the following way

$$P_i = \left(x\left(-\frac{\pi}{2} + i \cdot \frac{2\pi}{10} \right), y\left(-\frac{\pi}{2} + i \cdot \frac{2\pi}{10} \right) \right)^T, \tag{10}$$

for $i = 0, 1, \ldots, 10$. Starting with these control points we fit the Lemniscate of Gerono by the sequence of curves generated by Algorithm 1 using the uniform B-spline basis of degree 2 and the sequence $(t_i)_{i=0}^{m-3}$ considered previously for the particular case of $m = 13$ (dimension 11). In Figure 1 (a) we can see the curve which must be imitated and the starting control points extracted from this curve according to (10). In Figures 1 (b), (c) and (d), we can see the starting curve, and the curves of the sequence after 2 and 5 iterations obtained by Algorithm 1, respectively.

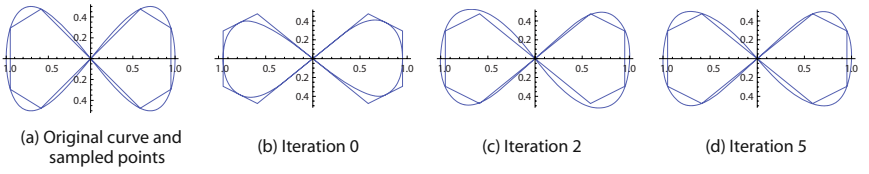


Fig. 1. Lemniscate of Gerono using the B-spline basis of degree 2

We can observe in these figures that Algorithm 1 using the uniform B-splines basis of degree 2 converges very quickly to the Lemniscate of Gerono. This fact can also be checked in Table 3 where we have shown the fitting errors of the sequence of curves after some specific iteration levels (the fitting error at the step $k \geq 0$ of Algorithm 1 is taken as the maximum Euclidean norm of the adjusting vectors $\Delta_0^k, \Delta_1^k, \ldots, \Delta_{10}^k$). As a proof of the efficiency of the algorithm we can observe in Table 3 that with approximately 20 iterations we get an error with the order of the single precision unit roundoff,

Table 3. Euclidean norm when fitting the Lemniscate using the B-splines basis of degree 2

Iteration	$\max_{0 \leq i \leq 10} \|\Delta_i^k\|_2$
0	$1.84602E-01$
5	$3.87977E-03$
10	$4.01472E-05$
15	$4.16318E-06$
20	$1.86870E-07$
25	$7.27520E-09$
30	$2.71943E-10$
40	$3.68590E-13$
50	$5.55112E-16$

whereas with approximately 50 iterations we get an error with the order of the double precision unit roundoff.

Now let us consider the helix of radius 5 given by the parametric equations

$$\begin{cases} x(t) = 5\cos t, \\ y(t) = 5\sin t \cdot \cos t, \\ z(t) = t, \end{cases} \tag{11}$$

$t \in [0, 6\pi]$. We took a sample of 19 points $\{P_i\}_{i=0}^{18}$ from the previous helix in the following way

$$P_i = \left(x\left(i \cdot \frac{6\pi}{18}\right), y\left(i \cdot \frac{6\pi}{18}\right), z\left(i \cdot \frac{6\pi}{18}\right) \right)^T, \tag{12}$$

for $i = 0, 1, \ldots, 18$. Starting with these control points we fit the helix by the sequence of curves generated by Algorithm 1 using the uniform B-spline basis of degree 2 and the sequence $(t_i)_{i=0}^{m-3}$ considered previously for the particular case of $m = 21$ (dimension 19). In Figure 2 (a) we can see the curve to be imitated and the starting control points extracted from this curve according to (12). In Figures 2 (b) and (c) we can see the starting curve and the curve of the sequence after 5 iterations obtained by Algorithm 1, respectively.

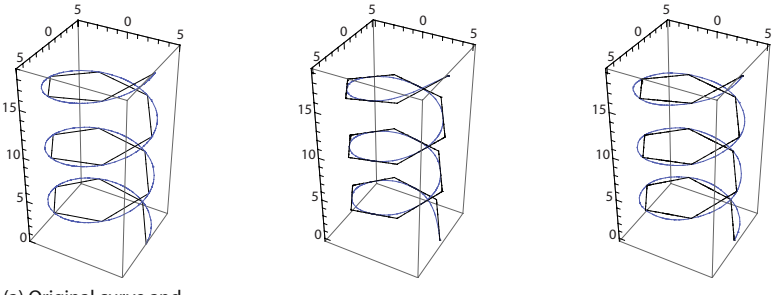


Fig. 2. Helix using the B-spline basis of degree 2

We can observe in these figures that Algorithm 1 using the uniform B-splines basis of degree 2 is very efficient for converging to the helix of radius 5. This fact can also be checked in Table 4 where we have shown the fitting errors of the sequence of curves after some specific iteration levels. As a proof of the fast convergence rate of the algorithm we can observe in Table 4 that with approximately 20 iterations we get an error with the order of the single precision unit roundoff, whereas with approximately 50 iterations we get an error with the order of the double precision unit roundoff.

Finally let us consider uniform B-spline bases of degree 3 over nonperiodic knot vectors $U = \{u_0, u_1, \ldots, u_m\}$ defined by (7) with $a = 0$ and $b = 1$. The dimension of the corresponding space of functions is $m - 3$. We will denote the B-spline basis obtained in this way by $(N_0^3, \ldots, N_{m-4}^3)$. Now, let us consider the strictly increasing sequence

Table 4. Euclidean norm when fitting the Helix using the B-splines basis of degree 2

Iteration	$\max_{0\leq i\leq 18}\|\Delta_i^k\|_2$
0	$2.27015E+00$
5	$3.03129E-02$
10	$5.45000E-04$
15	$1.27911E-05$
20	$3.46097E-07$
25	$1.01858E-08$
30	$3.18314E-10$
40	$3.52724E-13$
50	$8.88178E-16$

$(t_i)_{i=0}^{m-4}$ given by $t_i = i/(m-4)$. Then, we can apply Algorithm 1 with the basis and sequence previously considered.

For the case of the B-spline bases of degree 3 we have considered the same curves that in the case of the B-spline bases of degree 2, that is, the Lemniscate of Gerono given by (9) and the helix of radius 5 given by (11). Let us start with the Lemniscate of Gerono. We have taken the same sequence of 11 points $\{P_i\}_{i=0}^{10}$ from the Lemniscate of Gerono, given by (10) and (9), that were considered for the B-splines basis of degree 2. Starting with these control points we fit the Lemniscate of Gerono by the sequence of curves generated by Algorithm 1 using the uniform B-spline basis of degree 3 and the sequence $(t_i)_{i=0}^{m-4}$ considered previously for the particular case of $m = 14$ (dimension 11). In Figure 3 (a) we can see the curve which must be imitated and the starting control points extracted from this curve according to (10). In Figures 3 (b), (c) and (d), we can see the starting curve, and the curves of the sequence after 5 and 10 iterations obtained by Algorithm 1, respectively.

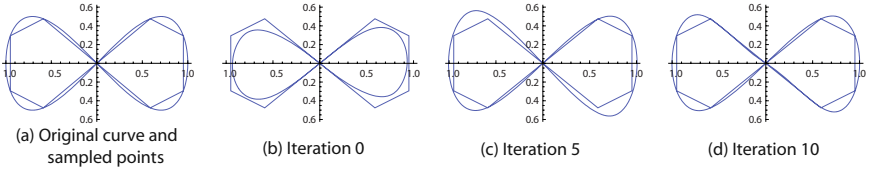


Fig. 3. Lemniscate of Gerono using the B-spline basis of degree 3

We can observe in these figures that Algorithm 1 using the uniform B-splines basis of degree 3 converges quite quickly to the Lemniscate of Gerono. This fact can be checked more precisely in Table 5 where we have shown the fitting errors of the sequence of curves after some specific iteration levels. In this case we need approximately 50 iterations in order to get an error with the order of the single precision unit roundoff. Obviously, Algorithm 1 using the B-spline bases of degree 3 is slower than the same algorithm using the B-spline bases of degree 2. We could expect this behaviour since the minimal eigenvalues of the corresponding collocation matrices of the B-spline

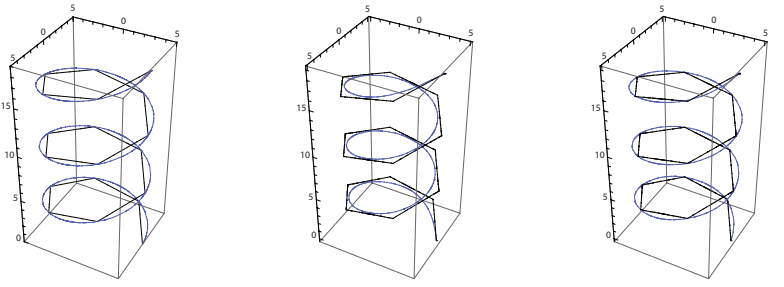


Fig. 4. Helix using the B-spline basis of degree 3

bases of degree 2 almost reach the minimal eigenvalue 0.5 of the collocation matrix of the Bernstein basis of degree 2 at the points $0, 1/2, 1$, while the minimal eigenvalue of the collocation matrix of the Bernstein basis of degree 3 at the points $0, 1/3, 2/3, 1$ is $2.22222E-01$ (see Tables 1 and 2). We have checked that the relationship between the minimal eigenvalues of the corresponding collocation matrices of the B-spline bases and the Bernstein basis of degree 3 is the same as that mentioned previously between the minimal eigenvalues of the collocation matrices of the Bernstein basis and the B-spline bases of degree 2.

Finally, in Figure 4 and Table 6 we can see the results obtained when applying Algorithm 1 using the corresponding B-spline basis of degree 3 to the helix of radius 5

Table 5. Euclidean norm when fitting the Lemniscate using the B-splines basis of degree 3

Iteration	$\max_{0\le i\le 10}\|\Delta_i^k\|_2$
0	$3.41935E-01$
10	$8.01351E-03$
20	$9.91843E-04$
30	$5.38469E-05$
40	$2.73960E-06$
50	$1.38409E-07$
60	$6.98716E-09$

Table 6. Euclidean norm when fitting the Helix using the B-splines basis of degree 3

Iteration	$\max_{0\le i\le 18}\|\Delta_i^k\|_2$
0	$3.79991E+00$
10	$1.52130E-01$
20	$9.14676E-03$
30	$7.70025E-04$
40	$6.29536E-05$
50	$4.83265E-06$
60	$3.56365E-07$

with the same 19 points given by (12) and (11). The conclusions are the same as those derived for the Lemniscate with the B-spline basis of degree 3.

4 PIA Algorithms Using Rational Bernstein Basis of Degree 2 and 3

In this section let us consider Algorithm 1 taking rational Bernstein bases of degree 2 and 3.

Let us start with a rational Bernstein basis of degree 2. Let us consider without loss of generality the rational Bernstein basis (r_0^2, r_1^2, r_2^2) given by

$$r_0^2 = \frac{b_0^2}{b_0^2 + w b_1^2 + b_2^2}, \quad r_1^2 = \frac{w b_1^2}{b_0^2 + w b_1^2 + b_2^2} \quad \text{and} \quad r_2^2 = \frac{b_2^2}{b_0^2 + w b_1^2 + b_2^2}.$$

Then, taking the increasing equally spaced sequence $(t_i)_{0 \le i \le 2}$ given by $t_i = i/n$ for $i = 0, 1, 2$ we get the collocation matrix

$$M\begin{pmatrix} r_0^2, r_1^2, r_2^2 \\ t_0, t_1, t_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2(1+w)} & \frac{w}{(1+w)} & \frac{1}{2(1+w)} \\ 0 & 0 & 1 \end{pmatrix},$$

whose eigenvalues are 1 (with multiplicity 2) and $\frac{w}{1+w}$, which is strictly increasing for all $w > 0$. Therefore, in order to get the optimal convergence rate of the PIA we must take w as high as possible, that is, $w = 2^{128}$ in single precision and $w = 2^{1024}$ in double precision. With those choices of the weight the eigenvalue $\frac{w}{1+w}$ is almost 1 and, hence, the method is very fast. Even with a considerably smaller w the corresponding convergence rate is higher than that of the Bernstein basis of degree 2 (see Table 1). Let us observe that high weights can produce overflow with the usual algorithm for evaluating rational Bézier curves. However, this problem is avoided when using the alternative algorithm proposed in [6], whose elementary steps are *all* convex combinations, in contrast to the usual rational algorithm.

Analogously, for degree 3 we can consider without loss of generality the rational Bernstein basis $(r_0^3, r_1^3, r_2^3, r_3^3)$ given by

$$r_0^3 = \frac{b_0^3}{b_0^3 + w b_1^3 + w b_2^3 + b_3^3}, \qquad r_1^3 = \frac{w b_1^3}{b_0^3 + w b_1^3 + w b_2^3 + b_3^3},$$
$$r_2^3 = \frac{w b_2^3}{b_0^3 + w b_1^3 + w b_2^3 + b_3^3} \quad \text{and} \quad r_3^3 = \frac{b_3^3}{b_0^3 + w b_1^3 + w b_2^3 + b_3^3}.$$

Then, taking the sequence $(t_i)_{0 \le i \le 3}$ given by $t_i = i/n$ for $i = 0, 1, 2, 3$ we get the collocation matrix

$$M\begin{pmatrix} r_0^3, r_1^3, r_2^3, r_3^3 \\ t_0, t_1, t_2, t_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{8}{9(1+2w)} & \frac{4w}{3(1+2w)} & \frac{2w}{3(1+2w)} & \frac{1}{9(1+2w)} \\ \frac{1}{9(1+2w)} & \frac{2w}{3(1+2w)} & \frac{4w}{3(1+2w)} & \frac{8}{9(1+2w)} \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

whose eigenvalues are 1 (with multiplicity 2), $\frac{2w}{1+2w}$ and $\frac{2w}{3+6w}$. The two last eigenvalues are strictly increasing for all $w > 0$. Therefore, in order to get the optimal convergence rate of the PIA we must take again w as high as possible, that is, $w = 2^{128}$ in single precision and $w = 2^{1024}$ in double precision. With these choices of the weight the eigenvalue $\frac{2w}{1+2w}$ is almost 1 whereas that $\frac{2w}{3+6w}$ is almost $\frac{1}{3}$. This last eigenvalue determines the convergence rate of the method, which is also fast in this case.

Remark 9. Let us observe that if we take w extremely high then the quality of the interpolant curve is affected. Hence we do not recommend taking values of w higher than 10. Nevertheless, with the choice $w = 10$ in the case $n = 2$ the minor eigenvalue of the corresponding collocation matrix is approximately 0.91, whereas in the case $n = 3$ it is approximately 0.32.

5 Another Approach of the Progressive Iteration Approximation Property

In Section 2 we have considered the iterative method corresponding to the progressive iteration approximation property as a method which gets a sequence of curves or surfaces represented with respect to the same basis but with different control points. In this section let us fix the points and let us modify the basis used for the curve or the surface representation.

Given a sequence of points $\{P_i\}_{i=0}^n$ whose i^{th} point is assigned to a parameter value t_i, for $i = 0, 1, \ldots, n$, and a blending basis $(u_0, \ldots, u_n)$, we can generate the starting curve of the PIA method as

$$\gamma^0(t) = \sum_{i=0}^{n} P_i u_i^0(t)$$

where $u_i^0 = u_i$ for all $i \in \{0, 1, \ldots, n\}$. Then, in order to compute the second curve of the sequence $\{\gamma^k\}_{k=0}^{\infty}$, we modify the used basis instead of modifying the control points. So we have

$$\gamma^1(t) = \sum_{i=0}^{n} P_i^1 u_i(t) = \sum_{i=0}^{n} P_i^1 u_i^0(t),$$

and then, taking into acount that

$$P_i^1 = P_i^0 + \Delta_i^0 = P_i + (P_i - \gamma^0(t_i)) = 2P_i - \sum_{j=0}^{n} P_j u_j^0(t_i),$$

we have that

$$\gamma^1(t) = \sum_{i=0}^{n} P_i 2u_i^0(t) - \sum_{i=0}^{n}\sum_{j=0}^{n} P_j u_j^0(t_i) u_i^0(t).$$

Changing the indices i and j in the second sum on the right hand side of the previous formula we can write

$$\gamma^1(t) = \sum_{i=0}^{n} P_i 2u_i^0(t) - \sum_{i=0}^{n}\sum_{j=0}^{n} P_i u_i^0(t_j) u_j^0(t) = \sum_{i=0}^{n} P_i \left[2u_i^0(t) - \sum_{j=0}^{n} u_i^0(t_j) u_j^0(t) \right].$$

Denoting $u_i^1(t) := 2u_i^0(t) - \sum_{j=0}^n u_i^0(t_j)\, u_j^0(t)$ for all $i = 0, 1, \ldots, n$ we have

$$\gamma^1(t) = \sum_{i=0}^{n} P_i\, u_i^1(t).$$

Iterating this process we get the following formula

$$\gamma^k(t) = \sum_{i=0}^{n} P_i\, u_i^k(t)$$

where $u_i^k(t) = (k+1)\, u_i^0(t) - \sum_{j=0}^n \sum_{r=0}^{k-1} u_i^r(t_j)\, u_j^0(t)$.

Let us consider the particular case where the basis $(u_0^n, \ldots, u_n^n)$ is an NTP basis of the space of polynomials of degree at most n. By Theorem 2 we know that $\lim_{k\to\infty} \gamma^k(t_i) = P_i$ for $i = 0, 1, \ldots, n$, that is, γ_k tends to the Lagrange interpolant of degree n as k tends to infinity. Taking into account that

$$P_i = \lim_{k\to\infty} \gamma^k(t_i) = \sum_{i=0}^{n} P_i \lim_{k\to\infty} u_j^k(t_i)$$

for $i = 0, 1, \ldots, n$, we can conclude that

$$\lim_{k\to\infty} u_j^k(t) = \frac{\prod_{i\neq j}(t - t_i)}{\prod_{i\neq j}(t_j - t_i)}$$

for $j = 0, 1, \ldots, n$, that is the Lagrange interpolation polynomials at $\{t_i\}_{i=0}^n$. In addition, if we choose the Bernstein basis of degree n, taking into account that it is the normalized B-basis of the space of polynomials of degree less than or equal to n, it provides, by Theorem 4, the fastest convergence to the Lagrange interpolation polynomials.

6 Conclusions and Future Work

This paper has analyzed the convergence rates of PIA for bases of different spaces. It is shown that B-spline bases presents advantages over Bernstein bases and that rational spaces can present the greatest advantages. The importance of an adequate choice of the weights and of an adequate evaluation algorithm has been also shown. The analysis of the NURBS case will be considered in a future work.

In particular we shall try to find bases for generating optimal NURBS, in the sense that the convergence rates of the PIA are maximal. Evaluation algorithms adequate for these bases will be studied.

Acknowledgements

The Spanish Research Grant MTM2009-07315, Gobierno de Aragón and Fondo Social Europeo have partially supported this research.

References

1. Ando, T.: Totally positive matrices. Linear Algebra and its Applications 90, 165–219 (1987)
2. Carnicer, J.M., Peña, J.M.: Shape preserving representations and optimality of the Bernstein basis. Advances in Computational Mathematics 1, 173–196 (1993)
3. Carnicer, J.M., Peña, J.M.: Totally positive bases for shape preserving curve design and optimality of B-splines. Computer Aided Geometric Design 11, 633–654 (1994)
4. Cooper, S., Waldron, S.: The eigenstructure of the Bernstein operator. Journal of Approximation Theory 105, 133–165 (2000)
5. de Boor, C.: How does agee's method work? In: Proceedings of the 1979 Army Numerical Analysis and Computers Conference, ARO Report 79-3, Army Research Office, pp. 299–302 (1979)
6. Delgado, J., Peña, J.M.: A corner cutting algorithm for evaluating rational Bézier surfaces and the optimal stability of the basis. SIAM Journal on Scientific Computing 29, 1668–1682 (2007)
7. Delgado, J., Peña, J.M.: Progressive iterative approximation and bases with the fastest convergence rates. Computer Aided Geometric Design 24, 10–18 (2007)
8. Horn, R.A., Johnson, C.R.: Topics in Matrix Analysis. Cambridge University Press, Cambridge (1994)
9. Lin, H., Wang, G., Dong, C.: Constructing iterative non-uniform B-spline curve and surface to fit data points (in chinese). Science in China (Series E) 33, 912–923 (2003)
10. Lin, H., Bao, H., Wang, G.: Totally positive bases and progressive iteration approximation. Computer & Mathematics with Applications 50, 575–586 (2005)
11. Peña, J.M.: Shape preserving representations in Computer Aided-Geometric Design. Nova Science Publishers, Inc. (1999)
12. Piegl, L., Tiller, W.: The NURBS Book, 2nd edn. Springer, Heidelberg (1997)
13. Qi, D., Tian, Z., Zhang, Y., Zheng, J.B.: The method of numeric polish in curve fitting (in chinese). Acta Mathematica Sinica 18, 173–184 (1975)

A Topological Lattice Refinement Descriptor for Subdivision Schemes

François Destelle, Cédric Gérot, and Annick Montanvert

Gipsa-Lab, CNRS-INPG, F-38000 Grenoble, France

Abstract. The topological step of a subdivision scheme can be described as a refinement on regular tiling lattices, or more generally as a local transformation descriptor. The former classifies all the regular lattice transformations which give rise to other regular lattices. These regular lattice descriptors are limited by the type of the control mesh faces, the subdivided mesh must be composed by faces of the same type. The latter category describes some local topological transformations as the insertion of vertices in each face, followed by the description of a connectivity; such a descriptor is called meta-scheme. But these meta-schemes cannot describe a large number of regular refinements. We propose a topological descriptor that generalizes meta-schemes. Our descriptor is locally defined by a triple of integers which describes the number of inserted vertices relatively to the components of each face: vertices, edges and the face center. It is combined with a flexible connectivity descriptor which enhances modeling capability. Our descriptor can build the schemes commonly used and it can describe a variety of others, including many regular rotative schemes. We ensure the conservation of the control mesh global topology. The subdivision operators described here can be concatenated, leading to more complex topological descriptions. In addition, we propose a general geometric smoothing step.

1 Introduction

Subdivision surfaces are a popular tool for the modelling of smooth surfaces from coarse meshes. Subdivision schemes are defined by the subdivided mesh topology coupled with the geometry of new vertices. This geometry is usually obtained by a linear combination of the old vertices *via* a topological correspondence between two subsequent lattices.

We introduce a lattice topological refinement descriptor. It describes the new faces of the refined mesh. This formalism allows a simple and intuitive encoding of the topological description of all usual subdivision schemes, as well as the creation of many new descriptions. The main characteristic of our descriptor is its generality: the faces of the control mesh may be of any type (triangles, quadrangles, hexangles or others). Thus we do not limit ourselves to the regular planar subdivision tilings, commonly used for subdivision surfaces due to the straightforward correspondence between two subsequent regular lattices. For a compact and general regular descriptor see [1]. For works about classical subdivision schemes see [2,3,4].

Our refinement descriptor is based on insertion of vertices into the faces of the control mesh. These inserted vertices are then connected with edges to build the faces of

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 153–177, 2010.

the refined mesh. Our complete description is composed by a vertex insertion descriptor and a connectivity descriptor. The word *meta-scheme* was introduced in [5] and [6] to describe potentially irregular subdivision descriptors. Our meta-scheme is a generalization of the existing ones. The description is *local* to a face of the control mesh and its immediate neighborhood. Any face can be subdivided by the same descriptor regardless of its type (triangle, quadrangle, etc.), and under some restriction, any face can be subdivided independently of the others. These choices allow the control mesh to be composed of several types of faces. The flexibility of our framework allows some interesting discussions, like the simple implementation framework proposed in appendix B.

First we summarize existing work on the classification of lattice refinements. In section 3 we describe our system, the insertion descriptor followed by the connectivity framework. We raise questions about local topological constraints of the refined meshes in subsection 3.3, we ensure that a refined mesh preserves the global topology of the initial one. In section 4 we show how the usual subdivision schemes and the irregular schemes presented in [5] and [6] can be described through our system. In subsection 4.3 we show that the concatenation of our descriptors can lead to more complex topological refinements. In 5 we expose the limitations of our framework. Finally, we propose a general geometric smoothing step in section 6.

2 Lattice Refinement Classifications

Several works on systematic classification exist for mesh subdivision. They display neither the same analysis framework nor the same limitations. We split them into regular classifications and possibly irregular ones.

On the first hand the word *regularity* comes from the regular tessellation of the Euclidean plane into triangles, quadrilaterals or hexagons. The valence of regular vertices are then 6, 4 and 3. In practice, such lattice refinements imply a restrictions: both the control and the refined meshes are composed of one and the same type of faces. In this case, the correspondence between two subsequent lattices is straightforward. In [7], the author considers regular triangular subdivision and introduces the main concept used here: a topological subdivision is an insertion of new vertices followed by a triangulation step. A simple and complete description encoding is presented in [1], it is based on a displacement vector between the two lattices: an edge of the control mesh projected onto the refined lattice is the scheme descriptor. The length of the vector represents the scheme arity and its orientation allows the introduction of rotative schemes, such as the topology of $\sqrt{5}$ in [8]. This description is composed by a two-dimensional vector and two letters which notify the type of the faces and the primality of the scheme; for example, $\mathrm{QP}(2,0)$ describes a Quadrangular Primal subdivision, and $\mathrm{TD}(0,3)$ is a Triangular Dual subdivision, see Figure 1. A large set of relevancy criteria concerning the usefulness of regular topological refinements for subdivision surfaces is introduced in [9]. The aim of the previous analysis is to ascertain whether there exist any potentially useful classes of topological refinement which have not yet been investigated.

On the other hand, the meta-schemes introduced in [5] and [6] present a local subdivision approach. They do not focus on a mapping between two unbounded regular lattices, but between a face of the control mesh and the resulting new faces of the refined mesh. These two works are based on the insertion of new elements in place of

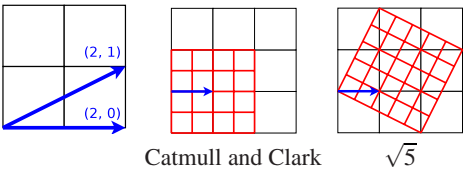


Fig. 1. Two examples of the regular topological subdivision descriptor [1]: QP$(2,0)$ and QP$(2,1)$ describe respectively the topology of Catmull and Clark scheme and $\sqrt{5}$ scheme

the vertices, the edges and the center of a face. The first one [5] considers three kinds of inserted elements, a vertex, an edge or a face is inserted instead of each kind of element. For instance, the notation $\mathscr{V}\mathscr{E}\mathscr{F} \rightarrow \mathscr{F}\mathscr{E}\mathscr{V}$ means that a face is created for each vertex, an edge is created for each edge and a vertex is created instead the center of the refined face. In addition, they provide an independent connectivity step to complete the description. The second one [6] considers only the insertion of vertices, while the possible connections are fixed and enumerated. It generalizes the former; it can describe all the refinements defined in [5] and more. In addition, they propose a solution to achieve a geometric smoothing, an inversion analysis of the subdivision process and they study the concatenation of several descriptors.

A regular descriptor cannot describe the irregular cases proposed by the meta-schemes, this description is constrained by the type of the faces; on the other hand, meta-schemes cannot describe, even by concatenation of several descriptors, some of the regular schemes when the arity is higher than 2, e.g. the topology of the ternary Loop scheme TP$(3,0)$. They are furthermore unable to describe most of the regular rotative schemes. Our system is a generalization of the meta-scheme [6]; we introduce more than one inserted vertex in place of each face element and we define a flexible and independent connectivity encoding. Among its description capabilities, we include a large number of regular schemes, in particular all schemes that verify the usefulness heuristics for subdivision surfaces [9]. Additionally, we can describe through our formalism all known meta-scheme refinements.

3 Framework of Our Meta-scheme

Our insertion descriptor $\mathscr{I}$ in a face $\mathscr{F}$ is a triple of integers: (n_v, n_e, n_f), the principle is the same as in [6]. It provides the number of inserted vertices relatively to the three kinds of elements in a face, see Figure 2. The connectivity descriptor $\mathscr{C}$ defines the new edges of the refined mesh; it is based on adjacency relations between the inserted vertices. A new face is then a minimal cycle in the refined mesh. The couple $(\mathscr{I}, \mathscr{C})$ defines without any ambiguity the topological subdivision of a face $\mathscr{F}$.

To be able to ignore the type of faces in a mesh, we define each operation in the same way for each edge in a face. As an example, the insertion of a vertex in place of an edge results in the insertion of a vertex for each edge in the face. Exploiting the rotational symmetry of a face, the insertion algorithm will describe vertices corresponding to a single half-edge of the face. We talk about meshes as half-edge structures. That is, data structures which contain oriented and linked edges into each face. Consequently,

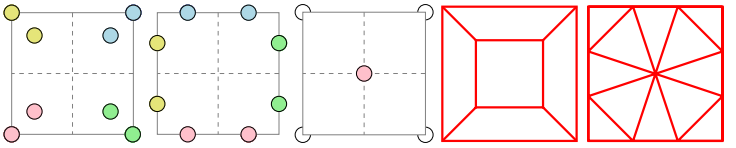


Fig. 2. Insertion of the three kinds of vertices: (a) vertices: $n_v = 2$, (b) edges: $n_e = 2$, and (c) center: $n_f = 1$. Some connectivity examples associated to the insertion triples (d) $(2,0,0)$, and (e) $(1,2,1)$.

we consider the *orientation* of faces, this feature allows the introduction of an order among the inserted elements. In the following, we consider every face as anti-clockwise oriented.

In subsection 3.3, we introduce a set of constraints to ensure that a refined mesh will preserve the initial topology of the control mesh. These constraints are based on the local topology of each graph created by the refinement of a face.

3.1 Insertion Descriptor $\mathscr{I}$

As mentioned above, the three integers n_v, n_e and n_f represent the number of vertices inserted in place of each kind of element: vertices, edges and the face center.

Insertion of vertices v_i

We insert n_v vertices for each vertex of the face $\mathscr{F}$, $n_v \in \mathbb{N}$. If $n_v = 1$, we insert a new vertex in place of each old vertex in $\mathscr{F}$. If $n_v > 1$, we insert and order $n_v - 1$ vertices for each vertex in $\mathscr{F}$.

For each vertex in $\mathscr{F}$, we consider an insertion sector, it is used as a two-dimensional discrete array. Each vertex v_i, $i \in [\![1, n_v - 1]\!]$ is located in this array by its coordinates (x_i, y_i). The insertion sector is related to each half-edge in the face, it is quadrilateral for all types of faces. We first define the half-edge insertion sectors considering the middle of edges and the face barycenter, see Figure 3. We need an order for the insertion and we need to fix the bounds (X,Y) of the array, $x_i \in [\![0, X-1]\!]$ and $y_i \in [\![0, Y-1]\!]$. We fill this array with successive vertices v_i column by column. The last column is not necessarily entirely filled, see Figure 3 right. The connectivity step is based on these half-edge insertion sectors and on the filling order of the inserted vertices, as explained in the next section. This spatial representation is not geometric, only the induced spatial adjacency relations are consistent. Our choices concerning the bounds (X,Y) of the array of vertices are justified by the regular quadrangular subdivision behavior and the local graph planarity constraint, see Figure 4.d. we provide a bound analysis in appendix A and we discuss the local graph planarity constraint in section 3.3. We consider the inferior integer value for the integer division operator in the following formula.

$$\begin{cases} Y = \lfloor \sqrt{n_v - 1} \rfloor \\ X = \lceil \frac{n_v - 1}{Y} \rceil \end{cases} \quad \text{and} \quad \begin{cases} x_i = i \ / \ Y \\ y_i = i \bmod Y \end{cases} . \tag{1}$$

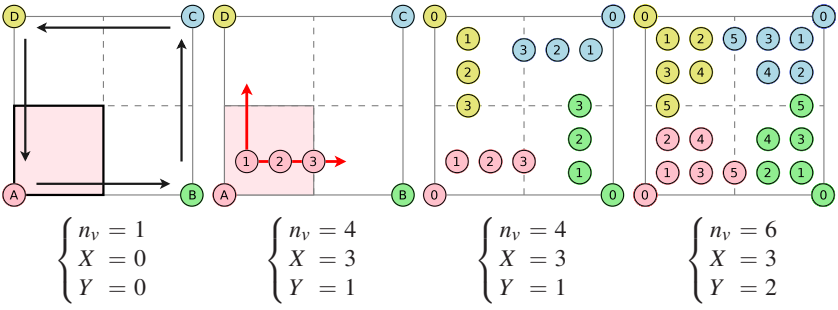


Fig. 3. Vertices v_i are ordered in an insertion zone related to each half-edge of the face, thus the process is the same for all types of faces. We denote these zone as half-edge insertion *sectors*. The connectivity step is based on these sectors and the order of the inserted vertices.

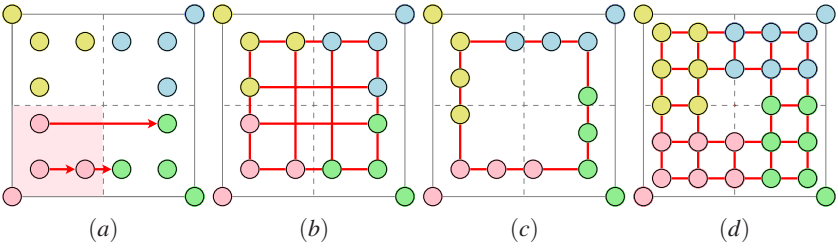


Fig. 4. (a) Descriptor $\mathscr{I} : (4,0,0)$ with $\mathscr{C} : v \rightarrow$ and $X = Y = 2$ for the vertices v_i. (b) These bounds induce a non-planar graph, some edges cross themselves. (c) Our bounds $X = 3$, $Y = 1$ prevent this situation for any descriptor $\mathscr{C} : v \uparrow\rightarrow$. (d) Example with $n_v = 7$, this configuration of vertices v_i can describe the regular scheme QP$(6,0)$.

Insertion of vertices e_i

We need to order the inserted vertices e_i, with $i \in [\![0, n_e - 1]\!]$ and $n_e \in \mathbb{N}$, along each edge. So they are uniformly spread over each edge, along the half-edge direction, see Figure 2.b.

The separation made between the description of vertices e_i and v_i provides much more capability to our formalism. As an example, all the regular triangular refinements require one more inserted edge vertex e_i than the number of inserted vertices v_i relative to each half-edge; for the quaternary Loop refinement TP$(4,0)$, we need three edge vertices e_i per half-edge and two vertices v_i. Moreover, all the regular rotative refinements require that we do not insert any vertex e_i, but the number of inserted vertices v_i can be greater than zero. The previous refinements described in our formalism are depicted in section 4.

Insertion of vertex f

We fix $n_f \in \{0,1\}$. If any, the vertex f is located at the barycenter of the old vertices in the face, see Figure 2.c. This choice avoids any ambiguity regarding the kinds of the inserted vertices in a face; notice that center of a face is the only rotationally invariant point. For the meta-scheme in [6] the related descriptors are $(n_v, n_e, n_f) \in \{0,1\}^3$.

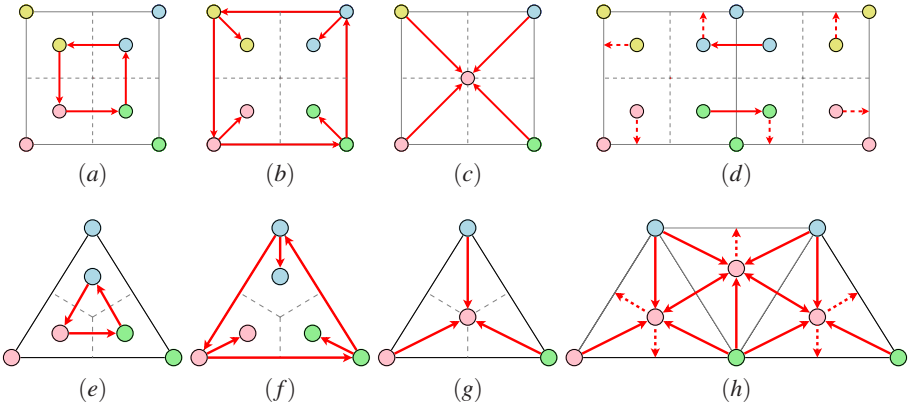


Fig. 5. We consider (a) and (e) the intra-element connectivities, here $v \rightarrow$ in $\mathscr{I} : (2,0,0)$, (b) and (f) the first vertex v_0 has two possible connections: $v_0 \rightarrow$ and $v_0 \nearrow$, (c) and (g) the inter-element connectivities, here $vf \uparrow$ in $\mathscr{I} : (1,0,1)$. The extern connectivities (d) create links between adjacent faces, here $\mathbf{v} \downarrow$ in $\mathscr{I} : (2,0,0)$. We can model the $\sqrt{3}$ scheme (h) with the connectivities $vf \uparrow$ and $\mathbf{f} \downarrow$ in $\mathscr{I} : (1,0,1)$).

3.2 Connectivity Descriptor $\mathscr{C}$

Our connectivity descriptor will induce the new edges of the refined mesh, linking the inserted vertices. It is defined by a set of symbols coupled with a set of directions represented by arrows. We need to link the different inserted vertices: vertices v_i are located into the face, vertices e_i are ordered along each edge and the vertex f (if any) is located at the face barycenter. Finally, the preserved old vertex v_0 will possess its particular connectivity. In addition, a set of local graph topology constraints are exposed in the next section; they ensure that a refined mesh will preserve the initial topology of the control mesh for any $(\mathscr{I},\mathscr{C})$.

Links between elements of the same kind are called *intra-element* connections, see Figure 5.a and Figure 5.b. The links between elements of different kinds, for instance between vertices v_i and f, are *inter-element* connections, see Figure 5.c. The previous links are said *intern* because the resulting new edges are located inside the old face.

The connections between elements in distinct faces are said *extern*. As an example, we will need an extern connection between the vertices f of adjacent faces to describe the topology of the $\sqrt{3}$ scheme [4], see Figure 5.h.

Vertices of the same kind are ordered, thus we can link them to each other in an intuitive manner. We need to set rules for the inter-element connectivity. Each inserted vertex of a same kind is processed the same way; as an example $\mathscr{C} : v \rightarrow$ will mean that all vertices v_i are linked to their right neighbor. The inter-element connection $\mathscr{C} : ev \uparrow$ will describe the connection of each vertex e_i to its upper neighbor v_i. In the following part, we will describe the different connections that are used throughout our framework, table 1 summarizes all the connection capabilities of our descriptor.

Table 1. The different directions of our connectivity descriptor. A link is Intern intra-element, Intern inter-element or Extern. There is no ambiguity among the notations but, to improve clarity, we employ a bold letter for the extern connectivity.

Intern intra-connections	Possibilities
v_0 v e	↗ → ↖ ↑ ↗ → ↑ →

Intern inter-connections	Possibilities
ev vf ef	← ↖ ↑ ↗ → ↑ → ↑

Extern connections	Possibilities
v **f**	↙ ↓ ↘ ↓

Intern intra-element connections

Connections of this class describe links between elements of the same kind, see Figure 5.a. They are noted by a single letter $\mathscr{C} : v$, e, or v_0 and a set of directions. The neighborhood of each vertex v_i is relative to the discrete domain introduced in the half-edge sectors. We consider the two immediate neighborhoods: $\uparrow$, $\rightarrow$, plus the diagonal directions $\nwarrow$ and $\nearrow$, see Figure 6. The neighbor of a vertex v_i along a direction is the first one reached by a discrete displacement from v_i along this direction. This neighborhood is extended to the next and previous half-edge sectors, see Figure 6.b. We note $|\mathscr{F}|$ the number of edges in $\mathscr{F}$. A half-edge sector $s \in [\![1, |\mathscr{F}|]\!]$ is adjacent to the two sectors s_{next} and s_{prev}, where $s_{next} = (s+1) \bmod |\mathscr{F}|$ and $s_{prev} = (s-1) \bmod |\mathscr{F}|$. Thus we have to consider two adjacent arrays of vertices v_i in our spatial neighborhood: $[X \times Y]$ and $[Y \times X]$, as the next sector array s_{next} is rotated by $\frac{\pi}{2}$ radians with respect to s. The first row of adjacent vertices v_i is composed of $X+Y$ vertices: the first row of the half-edge sector s plus the first column of the next sector s_{next}.

As it has been said, the first vertex v_0^s shows a difference, there are two different cases. v_0^s can be linked to the next sector first vertex $v_0^{s_{next}}$ *via* the connection $v_0 \rightarrow$. It can also be linked to v_1^s with the connection $v_0 \nearrow$, see Figure 5.b.

Two vertices e_i are neighbors if they are adjacent on the half-edge along the direction $\rightarrow$ ($e \leftarrow$ is redundant). When denoting $e \uparrow$, the last vertex e_i of a sector s is adjacent to the first vertex e_i of the next sector s_{next}. As an example, this connection can describe the topology of the Loop scheme, see Figure 7.a.

There is no intra-connection concerning the vertex f.

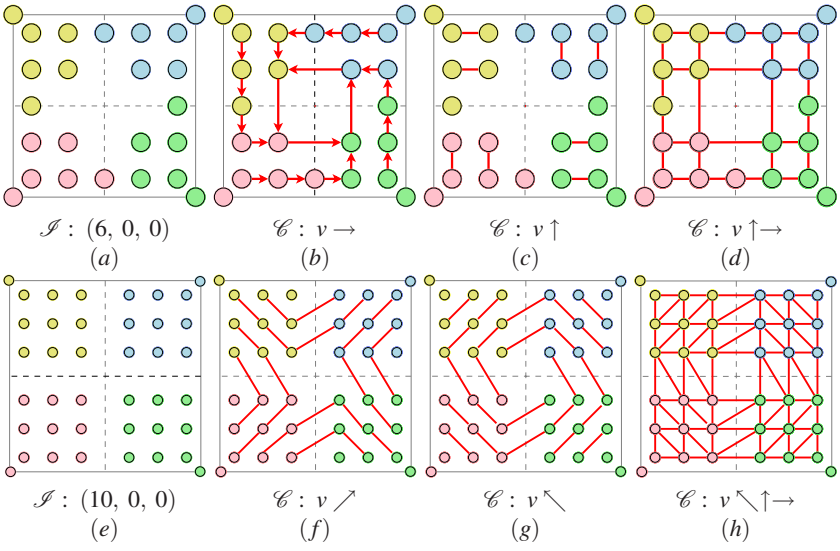


Fig. 6. Intern intra-element connectivity examples for vertices v_i. We define two connections at the same time in (d) and three in (h).

Intern inter-element connections

This connectivity builds the links between elements of different kinds, see Figure 5.c. We adopt the two-letter notations ev, vf and ef coupled with a set of directions. Some examples are depicted in Figure 7.

The first edge vertex e_0^s can be linked to v_0^s with the connection $ev \leftarrow$. Symmetrically the last vertex $e_{n_e-1}^s$ can be linked to the first vertex $v_0^{s_{next}}$ of the next sector with the connection $ev \rightarrow$.

The vertices e_i can be linked to the vertices v_i of the first row with the connections $ev \uparrow$, $ev \nwarrow$ and $ev \nearrow$. This first row contains the first row of the half-edge sector and the first column of the next sector: $v_j^s(x_j, 0)$ and $v_k^{s_{next}}(0, y_k)$, thus $X+Y$ vertices. On Figure 3, the example on the right shows five vertices on the first row for each half-edge sector. We consider two distinct cases; in an intuitive way, if $i < X$, a vertex e_i^s is linked to the vertex $v^s(i,0)$. If $i \geq X$, e_i is linked to the vertex $v^{s_{next}}(0, X+Y-(i+1))$ using the connection $ev \uparrow$, see Figure 7.d.

The connection $ef \uparrow$ links all the vertices e_i to the vertex f, see for example the topology of the ternary Loop scheme, Figure 7.c.

The two connections $vf \uparrow$ and $vf \rightarrow$ are not as intuitive as the former ones. The connection $vf \uparrow$ links the vertex f to v_0 or to the last vertex v_{n_v-1} if $n_v > 1$, see Figure 7.e. The connection $vf \rightarrow$ links the vertex f to the vertex $v(X-2, Y-1)$: it is located on the last row and the last but one column, see Figure 7.f. This direction is needed to triangulate the center domain of the refined face. It is used in particular to describe the regular triangular schemes shown later in subsection 4.2. These two connections allow us to join the distinct half-edge sectors.

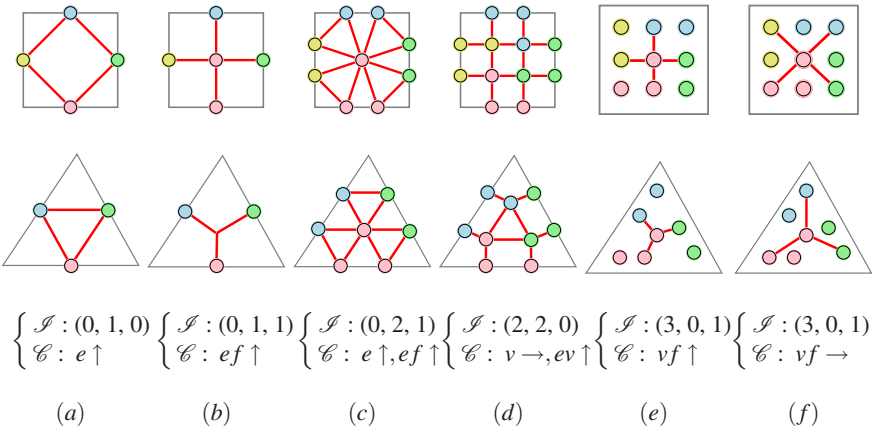


Fig. 7. The intra-element link $e\uparrow$ (a) and some intern inter-element connectivity examples applied on a quadrangle and a triangle: (b) and (c) $ef\uparrow$, (d) $ev\uparrow$, (e) $vf\uparrow$ and (f) $vf\rightarrow$. Vertices v_0 are not shown in (d), (e) and (f).

Extern connections

A connection is called extern if there is a link between two vertices in adjacent faces. Edge vertices e_i and the vertex v_0 are not concerned by the extern connectivity, since two adjacent faces share the same couple of half-edges. We denote an extern connection with one bold letter coupled with a direction. There is no ambiguity concerning intern and extern directions, we choose this notation to enhance clarity.

The face vertex f can be linked with adjacent vertex face centers, see Figure 8.a, this connection is noted by $\mathbf{f}\downarrow$. The vertices v_i in a half-edge sector can be linked to the vertices of the opposite half-edge sector with the the connection $\mathbf{v}\downarrow$, see Figure 8.c. This opposite domain is rotated by π radians with respect to the sector, but the adjacency process remains the same as before.

Finally, we define the two diagonal directions $\mathbf{v}\swarrow$ and $\mathbf{v}\searrow$. If a vertex v_i is not located on its domain borders ($x \neq 0$ and $y \neq 0$) these connections are ignored. This choice is motivated by the description of the regular quadrangular rotative tilings, shown in Figure 8.b and in section 4.2. A connection $\mathbf{v}\swarrow$ or $\mathbf{v}\searrow$ is needed but it may imply a triangulation of the vertices inside the half-edge sector.

3.3 Topology Constraints on the Refined Mesh

During the refinement process of a face, we aim to preserve the local topology of the mesh. A single face describes a two manifold, the graph is planar and there is only one connected component. A non-planar graph cannot be drawn in the plane without edge intersections. In this section we present rules to assure that a set of faces, created by the refinement of a face and its immediate neighborhood, remains a two-manifold description. These rules are necessary for a general topology preservation result. We split the analysis between the planarity of the new graph and its number of connected

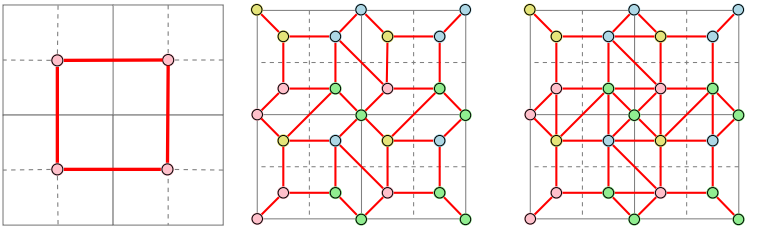


Fig. 8. The extern connections allow us to describe the dual mesh transformation and many regular rotative tiling refinements. (a) Dual mesh, (b) $\sqrt{5}$ scheme topology, and (c) $\sqrt{7}$ scheme topology, that is the regular triangular rotative scheme $\mathrm{TP}(2,1)$ applied on a quadrangular control mesh.

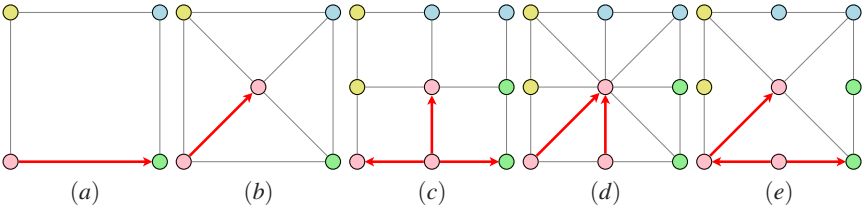


Fig. 9. Four examples of minimal connectivities, they are required to fulfill the topology constraints on the resulting graph. (a) $\mathscr{I} : (1,0,0)$, $\mathscr{C} : v_0 \rightarrow$, (b) $\mathscr{I} : (1,0,1)$, $\mathscr{C} : vf \uparrow$ (4-8 scheme topology). (b) and (c) are described by the same insertion descriptor $\mathscr{I} : (1,1,1)$ with three possible connectivities $\mathscr{C} : ev \leftarrow\rightarrow, ef \uparrow$, $\mathscr{C} : vf \uparrow, ef \uparrow$ and $\mathscr{C} : ev \leftarrow\rightarrow, vf \uparrow$.

components. Notice however that our framework can be equally applied to meshes that do not possess a two-manifold structure.

Local planarity

In order to ensure a locally refined planar graph, the following conditions must be fulfilled. We set four restrictions on intern intra-element connectivity:

- If two faces $\mathscr{F}_1$ and $\mathscr{F}_2$ share an edge (i.e. two opposite half-edges) they must specify the same descriptor concerning this edge. Thus n_e is the same for the two descriptors $\mathscr{I}_1$ and $\mathscr{I}_2$ and they must specify the same intra-element connections e.
- The connection $v \rightarrow$ links the first vertex v_0^s to the first one of the next sector $v_0^{s_{next}}$ only if there is no vertex e_i and if there is no extern connection.
- Intern connections $v \nwarrow$ and $v \nearrow$ cannot be specified simultaneously, see Figure 6.h.
- Intern connections $v_0 \nearrow$ and $e \uparrow$ cannot be specified simultaneously, see Figure 10.a.

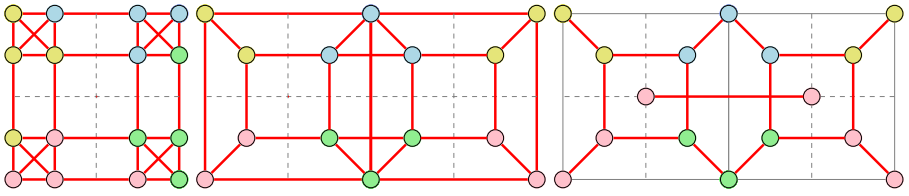


Fig. 10. A set of connectivities does not always give rise to a local planar graph

We set one restriction on intern inter-element connectivity:

- Connection $vf \rightarrow$ is allowed if vertices v_i respect the inequality: $X > Y$. That is if the length of the array columns is smaller than the length of its rows. For example, if $n_v = 10$ (thus v_0 and $X = Y = 3$ in each half-edge sector) see Figure 6.h, we cannot use this connection.

Finally we set three restrictions on extern connectivity:

- No extern connection is allowed if $\mathscr{C}$ contains one of the connections $v_0 \rightarrow$, $e \rightarrow$, $ev \rightarrow$ or $ev \leftarrow$, see Figure 10.b.
- Extern connections $\mathbf{v} \swarrow$ and $\mathbf{v} \searrow$ cannot be specified simultaneously.
- Connection $v \rightarrow$ and extern connections $\mathbf{v} \swarrow$ and $\mathbf{v} \searrow$ cannot be specified in addition to the extern connection $\mathbf{f} \downarrow$, see Figure 10.c.

Connectedness
To ensure the unicity of the connected component, we cannot leave any isolated vertex. Thus any kind of inserted element v, e or f must be associated with the same kind of connectivity. Some connections must be forced to ensure the creation of the new faces of the refined mesh. In Figure 9 are depicted several examples of minimal connectivities, they are mandatory to fulfill the local topological constraints.

Moreover, a sub-graph of the resulting mesh must not be isolated from the whole graph, like it is depicted in Figure 6.d, the new sub-graph inside the face is disconnected. For this particular case, a connection $v_0 \nearrow$ must be added. More generally, if $n_v > 1$ a connection $v_0 \nearrow$ or one of the three connections $ev \nwarrow \uparrow \nearrow$ must be added to fulfill the constraint.

3.4 Note on Dual Schemes

A refinement is *dual* if it is applied on the dual of the control mesh, noted $\mathscr{M}^*$. We describe the result of a topological dual subdivision as the dual mesh $\mathscr{M}'^*$ of the result of a primal subdivision $\mathscr{M}'$. The results $\mathscr{M}'^*$ and $\mathscr{M}^{*\prime}$ are similar except at the surface border if any. Thus, the dual subdivisions result from the composition of primal subdivisions with our description of the dual mesh. Our dual mesh insertion descriptor is $\mathscr{I} : (0,0,1)$, a face of the control mesh is considered as a vertex in the refined mesh. As an example, the Doo and Sabin scheme [10] is the dual Catmull and Clark scheme,

and the hexagonal scheme proposed by Dyn and al. [11] is the dual Loop scheme, topologically speaking. The composition of several descriptors is further developed in subsection 4.3.

4 Topological Refinements That Can Be Described

In addition to the defined descriptions, we can define a large set of topological refinements that cannot be described by the existing classifications. Notice that an application of a regular case descriptor of a kind on a different kind of face provides an irregular refinement. Thus our regular triangular description applied to quadrangles and pentagons is irregular, see Figure 11. Among all the possibilities, we can define a large number of irregular topological refinements *close* to the regular ones, topologically speaking. Some examples of such irregular refinements are illustrated in Figure 14. This example can be considered as a sequence of descriptors from a regular description into another. A slight variation in the insertion or in the connectivity descriptor results in a topologically close refinement.

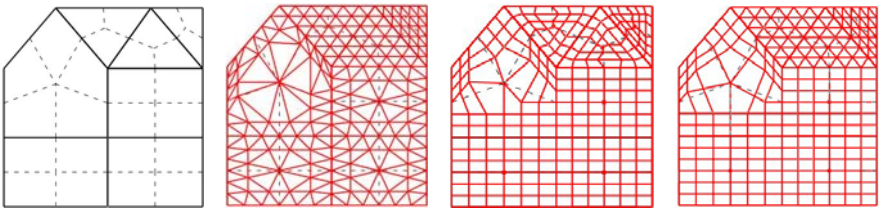


Fig. 11. Regular subdivision without rotation, $n_e = 5$, for (a) triangular, (b) quadrangular behavior. (c) An adaptative topological refinement example with regular triangular and quadrangular behaviors. The insertion triples are $\mathscr{I} : (4,5,1)$ and $\mathscr{I} : (7,5,1)$ for respectively the triangular and the quadrangular case.

4.1 Usual Subdivision Schemes

Here are shown some possibilities of our descriptor. First, in Table 2 we describe with our formalism the regular schemes [1] which are not rejected by the Dodgson heuristics [9]. We add the regular subdivisions TP$(4,0)$ and QP$(4,0)$ because the connectivity descriptors are the same for higher arities ($n \geq 4$), our connectivity descriptor does not necessarily increase its complexity as the arity of the regular schemes increases. These topological refinements are exposed in Figures 12 and 13. Second, we propose a method to describe through our descriptors a large number of regular refinements without rotation. Finally, we show in Table 3 the irregular refinements introduced by the meta-schemes [5] and [6] described through our formalism.

Table 2. Regular classifications [1]

Regular descriptor [1]	Usual name	Insertion $\mathscr{I}$	Connection $\mathscr{C}$
QP(1, 1)	4-8	(1, 0, 1)	$vf\uparrow$
QP(2, 0)	Catmull-Clark	(1, 1, 1)	$ve\longleftrightarrow$, $ef\uparrow$
QP(2, 1)	$\sqrt{5}$	(2, 0, 0)	$v_0\nearrow$, $v\uparrow$, $\mathbf{v}\swarrow$
QP(2, 2)	$\sqrt{8}$	(2, 1, 1)	$v_0\nearrow$, $ev\uparrow\nearrow$, $vf\uparrow$
QP(3, 0)	ternary	(2, 2, 0)	$\mathscr{C}_{\mathscr{Q}}$
QP(4, 0)	quaternary	(3, 3, 1)	$\mathscr{C}_{\mathscr{Q}}$
TP(1, 1)	$\sqrt{3}$	(1, 0, 1)	$vf\uparrow$, $\mathbf{f}\downarrow$
TP(2, 0)	Loop	(1, 1, 0)	$ev\longleftrightarrow$, $e\uparrow$
TP(2, 1)	$\sqrt{7}$	(2, 0, 0)	$v_0\nearrow$, $v\uparrow$, $\mathbf{v}\swarrow\downarrow$
TP(2, 2)	$\sqrt{12}$	(2, 1, 1)	$v_0\nearrow$, $ev\uparrow\nearrow$, $vf\uparrow$, $ef\uparrow$, $\mathbf{v}\downarrow$
TP(3, 0)	Loop ternary	(1, 2, 1)	$\mathscr{C}_{\mathscr{T}}$
TP(4, 0)	Loop quaternary	(2, 3, 0)	$\mathscr{C}_{\mathscr{T}}$
Not described	–	(2, 3, 1)	any $\mathscr{C}$

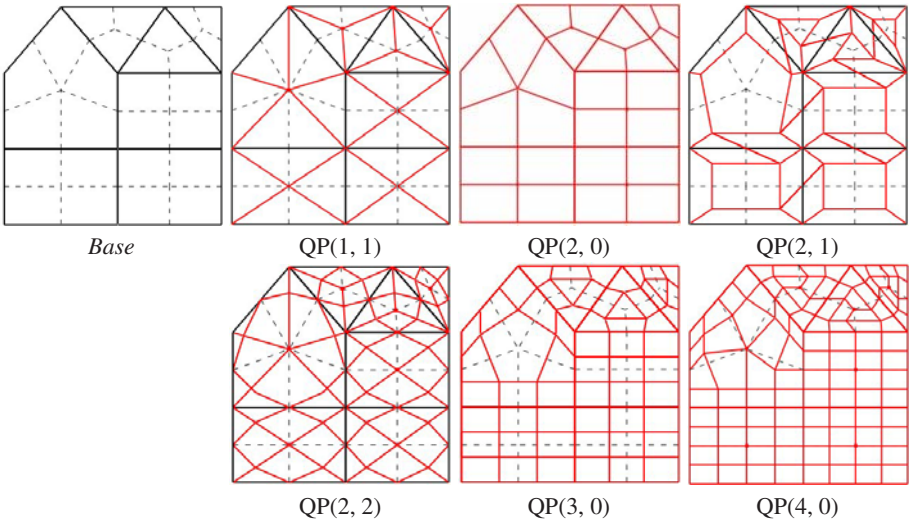


Fig. 12. Topological refinements similar to the primal quadrilateral schemes classified in [9], for a control mesh composed by a pentagon, three quadrangles and three triangles. These topological refinements are regular only if applied on quadrangles.

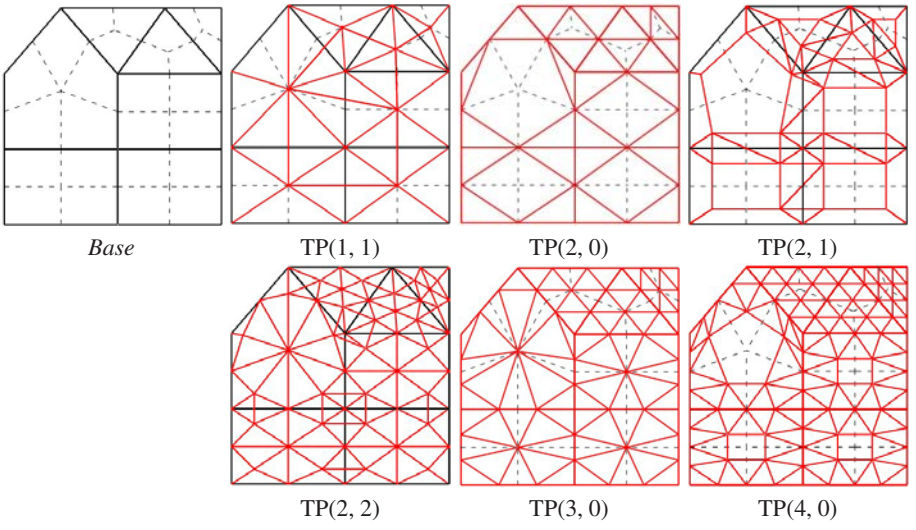


Fig. 13. The regular triangular refinement behavior. These topological refinements are regular only if applied on triangles.

Table 3. Meta-schemes [5] and [6]

Descriptor [5]	Descriptor [6]		Insertion $\mathscr{I}$	Connection $\mathscr{C}$
9	V	VV	(1, 0, 0)	$v_0 \rightarrow$
8	E	EE	(0, 1, 0)	$e \uparrow$
7	VE	VE	(1, 1, 0)	$ev \leftarrow\rightarrow$
7	VE	VE, EE	(1, 1, 0)	$ev \leftarrow\uparrow\rightarrow$, $e \uparrow$
5	VF	VF	(1, 0, 1)	$vf \uparrow$
3	VF	VF, VV	(1, 0, 1)	$v_0 \rightarrow$, $vf \uparrow$
2	EF	EF	(0, 1, 1)	$ef \uparrow$
2	EF	EF, EE	(0, 1, 1)	$e \uparrow$, $ef \uparrow$
1	VEF	VE, EF	(1, 1, 1)	$ev \leftarrow\rightarrow$, $ef \uparrow$
1	VEF	VF, EF	(1, 1, 1)	$vf \uparrow$, $ef \uparrow$
1	VEF	VE, VF	(1, 1, 1)	$ev \leftarrow\rightarrow$, $vf \uparrow$
1	VEF	VE,EF, VF	(1, 1, 1)	$ev \leftarrow\rightarrow$, $vf \uparrow$, $ef \uparrow$
Not described	Not described		(1, 2, 1)	any $\mathscr{C}$

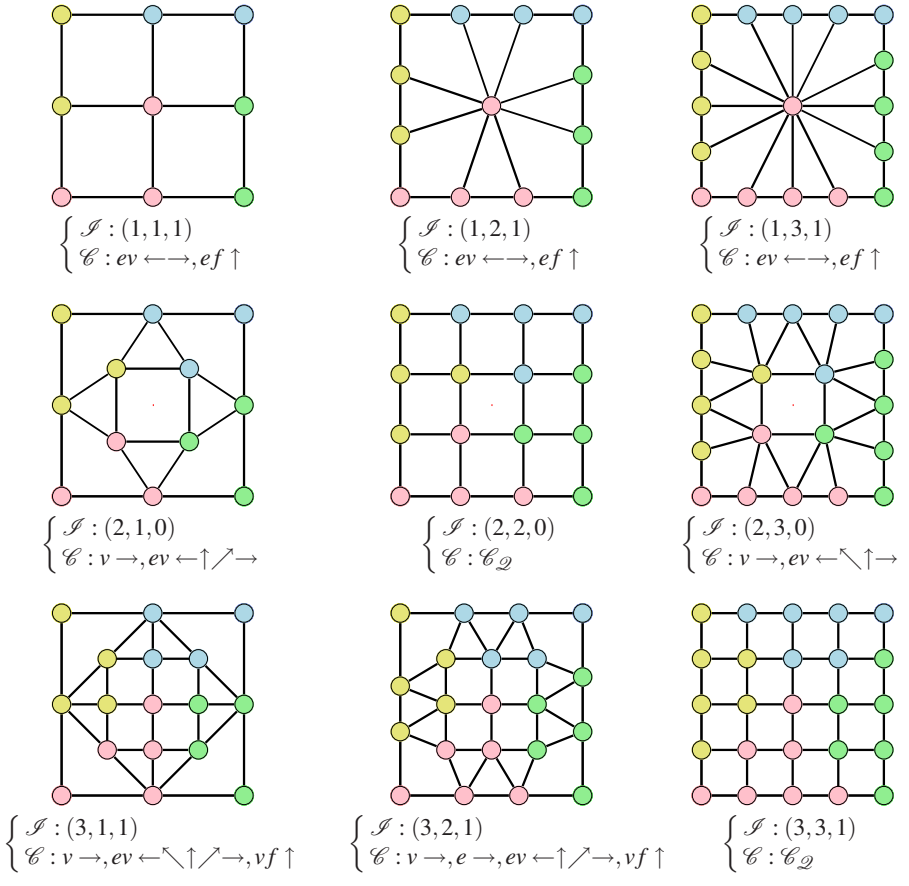


Fig. 14. Our descriptor can build a variety of non regular refinements. Here are shown several simple topological transitions between three regular refinements; we can imagine several intermediate descriptions between them.

4.2 Description of Regular Refinements without Rotation

In this section we describe how a regular subdivision without rotation can be described by our meta-scheme. Let the functions $\mathscr{R}_{\mathscr{Q}}$ and $\mathscr{R}_{\mathscr{T}}$ determine the insertion triple $\mathscr{I} : (n_v, n_e, n_f)$ from a face type and a chosen arity n, for a regular descriptor Q/TP$(n, 0)$. Beside, the connectivity descriptors $\mathscr{C}_Q$ and $\mathscr{C}_T$ define the connectivities of regular quadrangular and triangular behaviors without rotation.

$$\mathscr{R}_{\mathscr{Q}}(n) : \begin{cases} n_e = n-1 \\ n_v = 1 + \lfloor n_e^2/4 \rfloor \\ n_f = n_e^2 - 4(n_v - 1) \end{cases} \qquad \mathscr{R}_{\mathscr{T}}(n) : \begin{cases} n_e = n-1 \\ n_v = 1 + \lfloor \left(\frac{n_e(n_e-1)}{2}\right)/3 \rfloor \\ n_f = \lfloor \frac{n_e(n_e-1)}{2} \rfloor - 3(n_v - 1)\ . \end{cases} \tag{2}$$

$$\begin{cases} \mathscr{C}_{\mathscr{Q}} = v \rightarrow, \quad e \rightarrow, \quad ev \leftarrow \uparrow \rightarrow, \quad vf \uparrow \\ \mathscr{C}_{\mathscr{T}} = v \nwarrow \rightarrow, e \uparrow \rightarrow, ev \leftarrow \nwarrow \uparrow \rightarrow, vf \uparrow \rightarrow\ . \end{cases} \tag{3}$$

Some examples are depicted on Figure 11. On (c) we have adapted the connectivity descriptors locally accordingly to the type of faces, it is an adaptative topological refinement that can be associated to the topology of the subdivision scheme *Quad-Triangle* [12].

Not all regular refinements can be described by a descriptor couple $(\mathscr{I}, \mathscr{C})$. Beside this fact, many regular refinements can be described by the concatenation of two descriptors, this point is discussed in the next section. Refinements that cannot be described by our system are exposed in section 5.

4.3 Concatenation of Descriptors

New topological subdivision schemes can be built by concatenation of several simple descriptors. A concatenation is a successive application of several refinement descriptors $(\mathscr{I},\mathscr{C})$ on a face. In [6], the number of simple subdivision steps that are concatenated is called the *order* of the topological refinement description. The concatenation of all possible couples of descriptors lead to a very large number of topological transformations, the space spanned by the refined meshes created by our concatenated descriptors will be part of our future work; it requires a set theoretic study.

As mentioned in subsection 3.4, we can describe the dual topological refinements with the concatenation of a primal description and the dual descriptor $\mathscr{I} : (0,0,1)$.

We here present the special case of the description of the regular encoding [1] through the concatenation of two couples $(\mathscr{I},\mathscr{C})$. As an example, the concatenation of two Loop scheme topological descriptors TP$(2,0)$ produces the regular descriptor TP$(4,0)$. With our formalism, the preceding example is: $(\mathscr{I} : (1,1,0),\mathscr{C}_T) \oplus (\mathscr{I} : (1,1,0),\mathscr{C}_\mathscr{T}) = (\mathscr{I} : (2,3,0),\mathscr{C}_\mathscr{T})$. For simplicity, we use in the following the regular notations [1] rather than ours for the description of regular topological refinements.

For a regular scheme descriptor Q/TP(n,m) from [1], with $n,m \in \mathbb{N}^*$, we denote the arity and the rotation angle of the scheme respectively a and ϕ:

$$\begin{cases} a_\mathscr{Q} = \sqrt{n^2+m^2} & a_\mathscr{T} = \sqrt{(n+m/2)^2+(\sqrt{3}m/2)^2} \\ \phi_\mathscr{Q} = \arctan(m/n) & \phi_\mathscr{T} = \arctan(\frac{\sqrt{3}m/2}{n+\frac{m}{2}}) \; . \end{cases} \tag{4}$$

The result of the concatenation process is expressed through the arity and the rotation angle of each descriptor D_i. We advance the following formula

$$D_1(a_1,\phi_1) \oplus D_2(a_2,\phi_2) = D_3(a_1a_2, \phi_1+\phi_2) \; . \tag{5}$$

From there we go back to the notation Q/TP(n,m). Let be QP(n,m) = QP $(o,p) \oplus$ QP(q,r). The expression of the triangular case TP(n,m) follows the same reasoning (we do not detail it here). Since $\tan(x+y) = \frac{tan(x)+tan(y)}{1-tan(x)tan(y)}$ we have

$$\begin{cases} n^2+r^2 = (o^2+p^2)(q^2+r^2) \\ m/n = |\frac{pq+or}{oq-pr}| \text{ if } oq \neq pr \\ n = 0 \text{ otherwise.} \end{cases} \Rightarrow \begin{cases} n = |oq-pr| \\ m = pq+or \end{cases} \; . \tag{6}$$

Table 4. Examples of concatenations of regular primal descriptors. The right column is the result of the concatenation of the first and the second columns.

First	Second	$\oplus$
QP(o, p)	QP(q, r)	QP(oq-pr, pq+or)
QP(o, p)	QP(p, o)	QP(0, $p^2 + o^2$)
QP(o, p)	QP(x, 0)	QP(x.o, x.p)
QP(o, 0)	QP(1, 1)	QP(o, o)
QP(o, 0)	QP(2, 1)	QP(2o, o)

First	Second	$\oplus$
QP(2, 1)	QP(1, 1)	QP(1, 3)
QP(3, 1)	QP(1, 1)	QP(3, 4)
QP(2, 1)	QP(2, 1)	QP(3, 4)
QP(2, 2)	QP(2, 2)	QP(0, 8)
QP(3, 3)	QP(3, 3)	QP(0, 18)

The two symmetric definitions Q/TP(n,m) and Q/TP(m,n) are described by almost the same descriptors $(\mathscr{I},\mathscr{C})$, the difference is in the connection description which induces the tiling rotation: we use the links $\mathbf{v}\searrow$ and $\mathbf{v}\swarrow$ to model respectively the clockwise and the anti-clockwise rotations.

In our formalism, this regular case is written as follows. From equation 2, n_v and n_f can be determined from n_e.

$$\begin{gathered}(\mathscr{I} : (n_{v1}, n_{e1}, n_{f1}),\ \mathscr{C}_{\mathscr{Q}}) \oplus (\mathscr{I} : (n_{v2}, n_{e2}, n_{f2}),\ \mathscr{C}_{\mathscr{Q}}) = \\ (\mathscr{I} : (n_v, (n_{e1}+1)(n_{e2}+1)-1, n_f),\ \mathscr{C}_{\mathscr{Q}})\ .\end{gathered} \tag{7}$$

Table 5. Description of the concatenated refinements introduced in [5] and [6]

Descriptor [5]	Descriptor [6]		Insertion $\mathscr{I}$	Connection $\mathscr{C}$
12	VEF1/F	VE, EF	(1, 1, 1) $\oplus$ (0, 0, 1)	$ev \longleftrightarrow$, $ef \uparrow$
11	EF2/F	EF, EE	(2, 0, 0)	$v_0 \nearrow$, $v \rightarrow$
10	VF2/F	VF, VV	(1, 0, 1) $\oplus$ (0, 0, 1)	$v_0 \rightarrow$, $v \rightarrow$, $vf \uparrow$
5	E/EF2	EF, EE	(2, 0, 1)	$v \rightarrow$, $vf \uparrow$, $\mathbf{v}\downarrow$
6	VE2/F	VE, EE	(1, 1, 0) $\oplus$ (0, 0, 1)	$e \uparrow$, $ev \longleftrightarrow$
3	F/VF2	VF, VV	(1, 0, 1)	$vf \uparrow$, $\mathbf{f}\downarrow$
X	X	X	(2, 0, 1)	$vf \uparrow$, $\mathbf{f}\downarrow$

Table 4 is a brief summary of some regular quadrangular compositions denoted as in [1]. Our descriptors cannot describe the scheme QP$(4,2)$, which corresponds to $\mathscr{I}:(5,1,1)$, but this scheme can be expressed as an accessible composition: QP$(4,2) =$ QP$(2,0) \oplus$ QP $(2,1)$. The previous concatenation can be written with our notations

$$(\mathscr{I}:(1,1,1),\ \mathscr{C}:ev \longleftrightarrow, ef \uparrow) \oplus (\mathscr{I}:(2,0,0),\ \mathscr{C}:v_0 \nearrow \rightarrow, v \rightarrow, \mathbf{v} \swarrow)$$

Additionally, we show in Table 5 that our concatenated descriptors can describe all the concatenated refinements introduced in [6].

5 Limitations of Our Framework

Vertices v_i repartition
As exposed in subsection 3.1, our insertion framework for the vertices v_i is motivated by the regular quadrangular refinement behavior. We need to establish a correspondence between n_e vertices e_i and $X+Y$ vertices v_i. The regular triangulation must provide a correspondence between n_e vertices e_i and $X+Y-1$ vertices v_i. Thus we cannot model the regular schemes TP$(n,0)$ where $n>8$ without a concatenation of several descriptors. We can here imagine an automatic regular repartition of v_i relative to the type of the faces: triangles or other.

Regular refinements which cannot be described
From equation 6, we cannot describe the regular schemes Q/TP(n,m), $n,m>2$, that cannot be described as a concatenation of elementary operators; for the quadrangular case, we cannot describe QP(n,m) if there are no valid couples $(o,p),(q,r)$ as $n = oq-pr$ and $m = pq+or$. The schemes T/QP$(3,2)$ and T/QP$(2,3)$ are such examples. Furthermore, we cannot describe the regular *mixt* refinements MP(n,m), because we are constrained to process the connection step in the same way for all inserted elements of the same kind: v_0, v_i, e_i and f.

6 A Geometric Smoothing Step

As a final step, we propose to define a geometry on the refined mesh. It is an automatic framework designed to locate the inserted elements and to smooth the overall geometry. We cannot easily define a subdivision mask for each topological refinement and each kind of face, so we propose a method independent of the vertices valence and the type of faces. Our method is based on two geometric steps, we first locate our inserted vertices in each face of the control mesh, then we apply a low-pass filter on the geometry.

6.1 Mean Value Coordinates for Inserted Vertices v_i

First we need to express the geometry of the inserted vertices v_i as a convex combination of several old vertices in $\mathscr{M}$. This common geometric problem has been studied in [13], where a vertex location in a planar polygon is expressed by a convex combination of the vertices of this polygon.

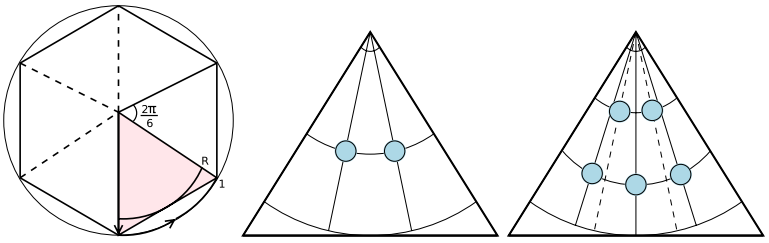


Fig. 15. A reference face is generated on the unit circle (a). We consider the central area for the vertex organization, $R = cos(\pi/|\mathscr{F}|)$. Then we order the vertices v_i with polar coordinates in a uniform manner, (b) two vertices: $n_v = 3$, (c) five vertices: $n_v = 6$.

These coordinates are known as *Wachspress* coordinates. Later, the *mean value* coordinates [14] were introduced. As the former ones, the resulting coordinates depend continuously and (infinitely) smoothly on the vertices of the polygon, but they are necessarily strictly positive.

For each refined face, we create a *reference* face, which is regular: each vertex $v_i(x_i, y_i)$ of a sector s is set in a uniform manner on the unit circle, see Figure 15. Then, for each edge s of the reference face, we uniformly order the inserted vertices v_i using polar coordinates (r, θ), see Figure 15.

$$\begin{cases} r_i = \frac{cos(\pi/|\mathscr{F}|)}{Y+2}(2+y_i) \\ \theta_i = \frac{2\pi}{|\mathscr{F}|}(s + \frac{1+x_i}{X+1}) \end{cases} \quad \begin{cases} \alpha_i = r_i sin\theta_i \\ \beta_i = r_i cos\theta_i \end{cases}. \tag{8}$$

As said earlier, the last row in the half-edge sector of the vertices v_i can be filled by $X-1$ vertices, then we replace X by $X-1$ in the preceding expression, see Figure 15.c; this particular case arises if $XY > n_v - 1$. The set of convex coefficients for each vertex v_i is then computed by the mean value method applied to the reference cartesian coordinate couple (α_i, β_i). We can now determine a set of geometric coefficients relative to a reference face of any type and apply it to any inserted vertices in the control mesh. This method has been studied for the coefficients of the subdivision scheme $\sqrt{5}$ in [8].

Before the geometric step, an inserted vertex f is located on the barycenter of the old vertices. The edge vertices e_i are ordered on the old half-edges in a uniform manner. A vertex e_i, $i \in [\![0, n_e - 1]\!]$, of an edge linking the two vertices V_a and V_b is defined by

$$e_i = \frac{(1+i)V_a + (n_e - i)V_b}{n_e + 1}. \tag{9}$$

6.2 Low-Pass Filtering of the Geometry

The mean value coefficients for the vertices v_i and the uniform vertices e_i and f distributions lead to planar refined faces. In order to use our subdivision schemes to model smooth surfaces, as in common subdivision schemes, we need to apply a local low-pass filter on the new vertices. We do not have any subdivision mask, so we use a geometric

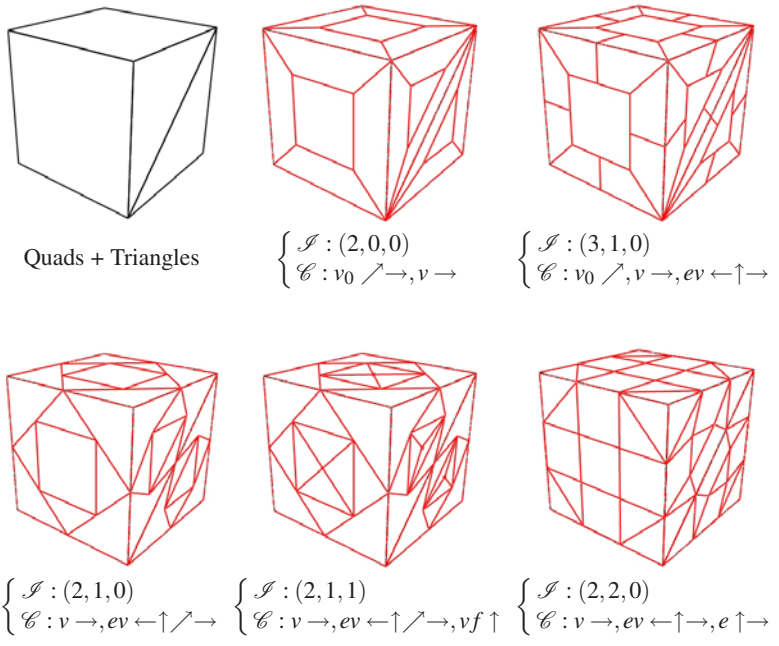


Fig. 16. Some surfaces build by our descriptors that can neither be described by the regular classification nor by the existent meta-schemes

heuristic. This step is highly tunable, at this point one can adjust the coefficients according to some fairness criteria. We choose to simply apply a mean filter on the vertex geometry. We consider their immediate neighbors on the refined mesh, instead of the old vertices of $\mathscr{M}$.

The new vertices v_0 replace the old vertices of $\mathscr{F}$. The new positions $\tilde{v_0}$ are given by half the sum of their old positions and the weighted sum of their immediate neighboring vertices, noted N

$$\tilde{v_0} = \frac{1}{2}\left(v_0 + \frac{1}{|N|}\sum_{k=1}^{k=|N|} N_k\right) . \tag{10}$$

For the new vertices e_i we consider their current position and the neighboring vertices which do not belong to the old edge, because they are set between two edge vertices e_i already. Each new position of vertex e_i is half the sum of their positions and the weighted sum of their geometric neighborhood which does not belong to the old edge, noted N^v

$$\tilde{e_i} = \frac{1}{2}\left(e_i + \frac{1}{|N^v|}\sum_{k=1}^{k=|N^v|} N_k^v\right) , \tag{11}$$

where N^v is the immediate neighborhood of a vertex e_i which contains neither another edge vertex e_i nor the vertices v_0^s and $v_0^{s_{next}}$.

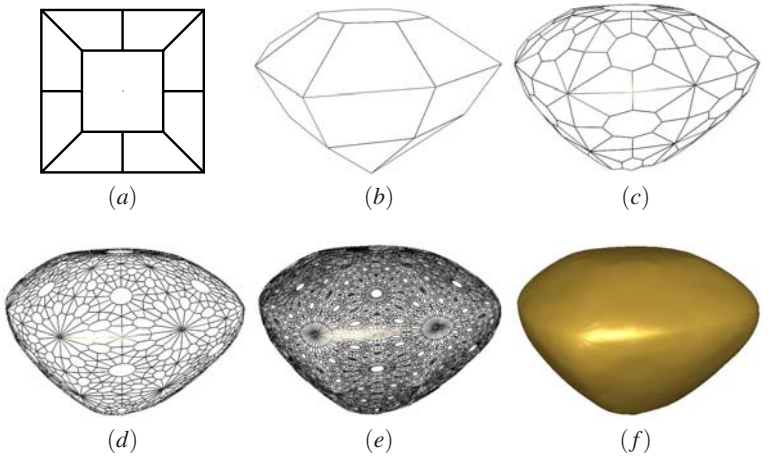


Fig. 17. (a) Our descriptor $\mathscr{I} : (3,1,0)$, $\mathscr{C} : v_0 \nearrow, v \rightarrow, ev \leftarrow\uparrow\rightarrow$ applied on a square. (b) A diamond-shape control mesh composed of an octagon, several quadrangles and several triangles. (c) One step of subdivision, (d) two steps, (e) three steps. (f) The enlighten limit shape.

The results show that the refined meshes are then quite smooth, see Figure 17. Of course, these subdivision schemes do not generate C^2 surfaces, but this geometric step can be applied to any of our topological refinement descriptions.

7 Conclusion

We have presented a construction framework that allows the description of a wide variety of topological refinements for subdivision surfaces. It extends meta-schemes with a description couple $(\mathscr{I},\mathscr{C})$. It can describe a large number of regular topological lattice refinements, in particular all of the useful topological subdivision schemes depicted in [9]. We have explored some of the adaptative topological refinements. We extend the description capabilities with the concatenation of two descriptors. We propose a simple but highly tunable general smoothing step for the refined meshes. Future works will focus on the study of the space spanned by the subdivision descriptors. We aim to achieve better geometric behaviors for our refined surfaces, and then be able to perform a fairness measure relative to each topological refinement.

References

1. Ivrissimtzis, I., Dodgson, N.A., Sabin, M.A.: A generative classification of mesh refinement rules with lattice transformations. Computer Aided Geometric Design 21(1), 99–109 (2004)
2. Loop, C.: Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah (1987)
3. Catmull, E., Clark, J.: Recursively generated B-spline surfaces on arbitrary topological meshes. Computer Aided Design 10(6), 350–355 (1978)

4. Kobbelt, L.: $\sqrt{3}$-subdivision. In: SIGGRAPH 2000 Conference Proceddings, pp. 103–112 (2000)
5. Kohler, M.: A meta scheme for iterative refinement of meshes. In: Hege, H.C., Polthier, K. (eds.) Mathematical Visualization, pp. 45–59. Springer, Heidelberg (1998)
6. Müller, H., Rips, M.: Another metascheme of subdivision surfaces. In: Hege, H.C., Polthier, K. (eds.) Visualization and Mathematics III, pp. 201–220. Springer, Heidelberg (2003)
7. Alexa, M.: Refinement operators for triangle meshes. Comput. Aided Geom. Des. 19(3), 169–172 (2002)
8. Ivrissimtzis, I., Dodgson, N., Sabin, M.: $\sqrt{5}$-subdivision. In: Dodgson, N., Floater, M., Sabin, M. (eds.) Advances in Multiresolution for Geometric Modelling, pp. 285–300. Srpinger, Berlin (2004)
9. Dodgson, N.A.: An heuristic analysis of the classification of bivariate subdivision schemes. In: IMA Conference on the Mathematics of Surfaces, pp. 161–183 (2005)
10. Doo, D.: A subdivision algorithm for smoothing down irregularly shaped polyhedrons. In: Proceedings on Interactive Techniques in Computer Aided Design, pp. 157–165 (1978)
11. Dyn, N., Levin, D., Liu, D.: Interpolating convexity-preserving subdivision schemes for curves and surfaces. Computer-Aided Design 24(4), 211–216 (1992)
12. Stam, J., Loop, C.: Quad/triangle subdivision. Graphics Forum 22(1), 79–85 (2002)
13. Wachspress, E.: A rational finite element basis. Mathematics in Science and Engineering (1975)
14. Floater, M.S.: Mean value coordinates. Comput. Aided Geom. Des. 20(1), 19–27 (2003)

Appendix

A Horizontal X and Vertical Y Bounds on the Spatial Domain for the Vertices v_i

We propose a definition of the bounds (X,Y) of the half-edge insertion sectors for which any insertion descriptor $\mathscr{I}$ coupled with any valid connection descriptor $\mathscr{C}$ gives rise to locally planar graph, see section 3.3. The array of inserted vertices $v_i(x_i,y_i)$, $i \in [\![1, n_v - 1]\!]$ is defined by the two bounds X and Y and by a filling order. Thus $x_i \in [\![0, X-1]\!]$ and $y_i \in [\![0, Y-1]\!]$. We choose to insert the vertices v_i in the array column by column. The first one v_1 is set at $(0,0)$; if $X > 1$ the second one if v_2 is set at $(0,1)$, $(1,0)$ otherwise, and so on. Some array bounds are not useful in practice, they give rise to local non-planar graphs when the connectivity descriptor is applied, see Figure 4.b: some edges cross themselves. Additionally, we aim to easily describe the regular refinement schemes through our formalism.

First, to avoid the crossing of edges, the last line of the array must not contains less vertices than there are in a filled column, see Figure 4.a. Second, the arrangement of vertices v_i should make possible the integration of most regular refinements. As exposed earlier, we have to link n_e vertices e_i with $X+Y$ vertices v_i. For the regular quadrangular case this implies $n_e = X+Y$. For the regular triangular case this implies $n_e = X+Y+1$; we need to choose between these two constraints on X and Y. We choose to fulfill the quadrangular constraint, the regular triangular case is easily integrated with these two

bounds for regular refinements up to arity 8 (without tiling rotation). Finally, we search for the minimal values of (X,Y) that verify these conditions.

Regular schemes QP$(n,0)$ with n odd
We have $n_e = \sqrt{4(n_v-1)} = 2\sqrt{n_v-1}$ (from equation 2 in section 4.2), which is an integer. We note that $\sqrt{n_v-1} = \frac{n_e}{2}$ is an integer because if n is odd, n_e is even. Consider the system

$$\begin{cases} X+Y = 2\sqrt{n_v-1} \\ X \ . \ Y \geq \ n_v-1 \ , \end{cases} \tag{12}$$

the minimal values of X and Y solutions of this system are

$$\begin{cases} X = \sqrt{n_v-1} \\ Y = \sqrt{n_v-1} \ . \end{cases} \tag{13}$$

Regular Schemes QP$(n,0)$ with n Even: We have $n_e = \sqrt{4(n_v-1)+1}$. We can see that

$$\begin{cases} & \sqrt{4(n_v-1)+1} \in \mathbb{N} & \\ \Rightarrow & \lfloor \sqrt{4(n_v-1)} \rfloor & = \sqrt{4(n_v-1)+1}-1 \\ \Rightarrow & \lfloor \sqrt{4(n_v-1)} \rfloor & = n_e-1 \\ \Rightarrow & n_e & = 2\lfloor \sqrt{n_v-1} \rfloor + 1 \ . \end{cases} \tag{14}$$

So we have

$$\begin{cases} X+Y = 2\lfloor \sqrt{n_v-1} \rfloor + 1 \\ X \ . \ Y \geq n_v-1 \ . \end{cases} \tag{15}$$

The solution couple of minimal values for the regular bound setting for all QP$(n,0)$ is

$$\begin{cases} X = \lceil \ \frac{n_v-1}{Y} \ \rceil \\ Y = \lfloor \ \sqrt{n_v-1} \ \rfloor \ . \end{cases} \tag{16}$$

Moreover, this expression implies that the last row of the domain array contains at least the number of vertices v_i there is in a filled column. The last row of a domain contains $\lfloor \frac{n_v-1}{Y} \rfloor$ vertices, we have $\forall n_v \in \mathbb{N}^*$: $Y \leq \lfloor \sqrt{n_v-1} \rfloor$, thus $Y \leq \lfloor \frac{n_v-1}{Y} \rfloor$ and our planar graph constraint is also fulfilled. It can be proved that the regular triangular constraint $n_e = X+Y+1$ is fulfilled for refinements of arity up to 8.

B A Simple Implementation Model

We present here a simple implementation model for our topological refinement descriptor. This algorithm builds the new faces $\mathscr{F}'_i \in \mathscr{M}'$ induced by a refinement of $\mathscr{F} \in \mathscr{M}$ described by a description couple $(\mathscr{I}, \mathscr{C})$. By construction, the mesh is designed to be a half-edge structure. We assume that adjacency procedures are known, they are defined in subsection 3.2. Moreover, we assume that all the constraints on $(\mathscr{I}, \mathscr{C})$ are stored in a list structure, thus we have a procedure to test if a refinement descriptor is valid: the connectivity is at least minimal and it respects the local graph planarity constraint. Notice that if a description couple $(\mathscr{I}, \mathscr{C})$ is valid, all the cycles of the refined sub-graph

are minimal. We naturally split our algorithm between three subsequent creation steps: vertices, connectivity and faces $\mathscr{F}'_i$.

Algorithm 1. Creation of new faces $\mathscr{F}'_i$ from $(\mathscr{I}, \mathscr{C})$

Require: $(\mathscr{I} : (n_v, n_e, n_f), \mathscr{C})$ is valid
  {*Initialization*}
  $\mathscr{F}'_i \leftarrow \{\emptyset\}$
  list_vertices $\leftarrow \{\emptyset\}$
  list_halfedges $\leftarrow \{\emptyset, \emptyset\}$
  $Y \leftarrow \lfloor \sqrt{n_v - 1} \rfloor$
  $X \leftarrow \lceil \frac{n_v - 1}{Y} \rceil$
  count $\leftarrow 0$

  {*Creation of vertices from* $\mathscr{I}$}
  if $n_f > 0$ **then**
    list_vertices $\leftarrow f$
  end if
  for all half-edges $e_i \in \mathscr{F}$ **do**
    if $n_v > 0$ **then**
      list_vertices $\leftarrow v_0^i$
      for $j = 1$ to $n_v - 1$ **do**
        list_vertices $\leftarrow v_j^i(j/Y, j \bmod Y)$
      end for
    end if

    if $n_e > 0$ **then**
      for $j = 0$ to $n_e - 1$ **do**
        list_vertices $\leftarrow e_j^i$
      end for
    end if
  end for

  {*Creation of half-edges from* $\mathscr{C}$}
  for all vertices $v_i \in$ list_vertices **do**
    for all connectivity $c \in \mathscr{C}$ of the same kind **do**
      neighbor $\leftarrow$ Adjacency(v_i, c)

      {Half-edge structure: next and opposite}
      if neighbor $\neq \{\emptyset\}$ **then**
        list_halfedges $\leftarrow he_{[2count+0]}(v_i$, neighbor$)$
        list_halfedges $\leftarrow he_{[2count+1]}($neighbor $, v_i)$
        previous $\leftarrow$ Find(list_halfedges, any, v_i)
        Next($he_{[2count+0]}$) $\leftarrow$ previous
        previous $\leftarrow$ Find(list_halfedges, any, neighbor)

```
        Next(he_[2count+1]) ← previous
        count ← count + 2 {Next one}
      end if
    end for
end for

{Creation of F'_i, minimal cycles in the new graph}
count ← 0
for all new half-edges he_i do
  if he_i ∉ F'_i then
    face ← {∅}
    next ← Next(he_i)
    while next ≠ he_i do
      face ← next
      next ← Next(next)
    end while
    F'_i ← face
  end if
end for
```

Subdivision Schemes and Norms

Serge Dubuc

Département de mathématiques et de statistique, Université de Montréal,
C.P. 6128, Succ. Centre-ville, Montréal (Québec), Canada H3C 3J7
dubucs@dms.umontreal.ca
http://www.dms.umontreal.ca/Professeurs/dubucs/index.html

Abstract. We compare three criteria of convergence of subdivision sche-mes for curves. The first involves the contraction of the differences of the refinements, the second, the contraction of the modulus of continuity of the refinements. The third one resorts to the notion of joint norm and may be used only for affine periodic subdivision schemes. We will relate these three criteria together by using specific norms for subdivision operators which make them equivalent for their respective scope. The duality theory plays an important role.

1 Introduction

A *(binary) subdivision matrix* is a matrix $S = (s(i,j) : i,j \in \mathbb{Z})$ for which there exists an interval $[\sigma,\sigma']$ such that $s(i,j) = 0$ if $i - 2j \notin [\sigma,\sigma']$. The interval $[\sigma,\sigma']$ is a *support*. A subdivision matrix generates a *subdivision scheme* $\mathscr{S}$. The initial state of the scheme is a function $f_0 : \mathbb{Z} \to \mathbb{R}$. The sequence of *refinements* of the scheme, $f_n : \mathbb{Z} \to \mathbb{R}$, $n \geq 0$, is recursively defined by

$$f_{n+1}(i) = \sum_{j\in\mathbb{Z}} s(i,j) f_n(j),\ i \in \mathbb{Z}, n \geq 0.$$

As example, we consider a subdivision scheme whose refinements are given by quadratic interpolation centred around even integers (see Fig. 1). For $n = 0,1,2,\ldots$ and $k \in \mathbb{Z}$, we refine according to the rule

$$f_{n+1}(4k+i) = p(2k+i/2), \quad i = 0,\pm1,\pm2$$

if p is the quadratic polynomial such that $p(2k+i) = f_n(2k+i), i = 0,\pm1$.

The corresponding subdivision matrix is

$$S = \begin{pmatrix} \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & 1 & 0 & 0 & \cdots \\ \cdots & 3/8 & 3/4 & -1/8 & \cdots \\ \cdots & 0 & \mathbf{1} & 0 & \cdots \\ \cdots & -1/8 & 3/4 & 3/8 & \cdots \\ \cdots & 0 & 0 & 1 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix}. \tag{1}$$

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 178–193, 2010.

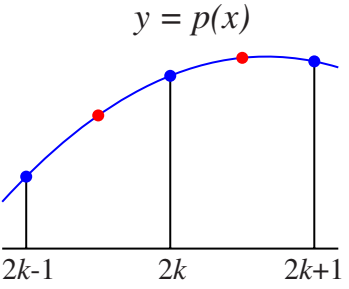


Fig. 1. Refinement by quadratic interpolation around even integers

The entry $s(i,j)$ in bold corresponds to the rank $i=0$ and the column $j=0$. If $i \in \{-2,-1,0,1,2\}$ and $s(i,j) \neq 0$, then $j \in \{-1,0,1\}$. For all $i,j \in \mathbb{Z}$, $s(i+4,j+2) = s(i,j)$.

We begin by recalling some definitions. A subdivision scheme $\mathscr{S}$ is C^0 *convergent*, or simply C^0 if, for every sequence of refinements $f_n : \mathbb{Z} \to \mathbb{R}$, the sequence of polygonal lines $\{(i/2^n, f_n(i)) : i \in \mathbb{Z}\}$ converges uniformly in any finite interval to the graph of $\phi : \mathbb{R} \to \mathbb{R}$. ϕ is called the *limit* of the refinements f_n. In Fig. 2, we get the limit function of the refinements of the subdivision scheme by quadratic interpolation where $f_0(0) = 1, f_0(1) = 1/2$, otherwise $f_0(k) = 0$.

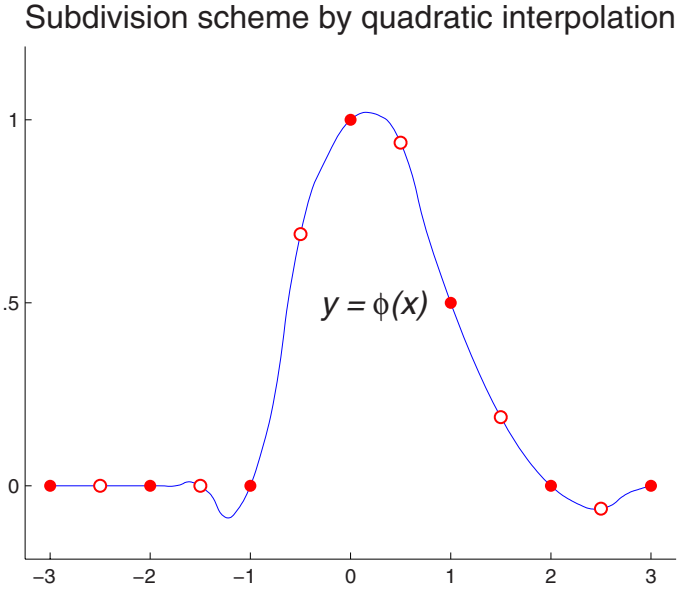


Fig. 2. One refinement and the limit of the refinements of a subdivision scheme by quadratic interpolation: • : f_0 ○ : f_1 —— : the limit ϕ

A subdivision matrix S is *affine* if $S\mathbf{1} = \mathbf{1}$, i.e. for all $i \in \mathbb{Z}$, $\sum_j s(i,j) = 1$. It is *periodic of period* p if

$$s(i+2p, j+p) = s(i,j) \quad \text{for all } i,j \in \mathbb{Z}.$$

The subdivision matrix (1) is periodic of period 2. A subdivision matrix S is *uniform* if it is periodic of period 1. A subdivision scheme is affine, periodic or uniform if the corresponding subdivision matrix has the respective property.

Classically, a subdivision scheme is given in terms of its subdivision mask, a function $a : \mathbb{Z} \to \mathbb{R}$. In this situation, the subdivision matrix whose entries are $s(i,j) = a(i-2j)$ is uniform i.e. periodic of period 1.

The main objective of this paper is to propose and compare three criteria of C^0 convergence of subdivision schemes for curves. They all involve norms of subdivision matrices. In Sect. 2, we investigate the contractivity of the differences of a subdivision scheme, what is the root of the first criterion. In Sect. 3, we see how to apply this criterion by computing the difference norm of S^n. There are two distinct ways to do that, as a primal problem of optimization or as a dual one. In Sect. 4, we validate a second criterion that involves the contractivity of the modulus of continuity of the refinements. The first two criteria are equivalent. In Sect. 5, we state the third criterion which applies only to affine periodic subdivision schemes. We show that the last two criteria are equivalent for periodic schemes. In Sect. 6, by numerical computations, we compare these three criteria for the subdivision scheme defined by quadratic interpolation. Some conclusions follow in the last section.

Particular instances of the first criterion have been found by Gregory, Dyn and Levin [6] in the uniform case and by Buhmann and Micchelli [1] in the periodic case. The third criterion extends a previous result of Daubechies and Lagarias [4] connected with uniform schemes. These criteria are closely related to the notion of spectral radius. The theme of spectral radius is recurrent in the field of subdivision schemes, and such occurrences happen in the works of Jia and Jiang [7] and Charina, Conti and Sauer [3]. The same last three authors, in their other paper [2], consider a criterion of convergence based on specific norms for subdivision schemes quite similar to our first criterion. What is different in their case is that it applies to uniform schemes which may be multivariate.

2 Contraction of Differences

The main tool of our analysis is the normed space $\ell_0(\mathbb{Z}) = \{f : \mathbb{Z} \to \mathbb{R} : \{i : f(i) \neq 0\} \text{ is finite}\}$ with the *difference norm*

$$\delta(f) = \max\{|\Delta f(i)| : i \in \mathbb{Z}\}$$

where Δ is the forward difference operator, $\Delta f(i) = f(i+1) - f(i)$. Any subdivision matrix $(s(i,j))$ induces an operator on $\ell_0(\mathbb{Z})$:

$$Sf(i) = \sum_{j \in \mathbb{Z}} s(i,j) f(j).$$

The *norm of S with respect to δ* is defined as $\delta(S) = \sup\{\delta(Sf) : \delta(f) \leq 1\}$.

Lemma 1. *A subdivision operator S on $\ell_0(\mathbb{Z})$ is bounded with respect to the difference norm δ if and only if the sum of every rank $\sum_{j\in\mathbb{Z}} s(i,j)$ is the same and the entries $s(i,j)$ are uniformly bounded.*

Proof. Let $[\sigma,\sigma']$ be the support of S. In the first part of the proof, we assume that the subdivision operator S is bounded with respect to the difference norm δ. We set $M=\delta(S)$.

Let $i\in\mathbb{Z}$, we choose an interval $[a,b]$, $a,b\in\mathbb{Z}$, such that $s(i,j)=s(i+1,j)=0$ for all $j\notin[a,b]$. For any integer N, we define the piecewise linear function f such that $f(a)=f(b)=1$, $f(a-N)=f(b+N)=0$, $\Delta^2 f(j-1)=0$ if $j\notin\{a-N,a,b,b+N\}$. Let $g=Sf$, then

$$g(i+1)-g(i)=\sum_{j\in\mathbb{Z}}[s(i+1,j)-s(i,j)]f(j)=\sum_{j=a}^{b}[s(i+1,j)-s(i,j)].$$

Since for all j, $|\Delta f(j)|\le 1/N$, then $\delta(f)\le 1/N$ and $|\Delta g(i)|\le M/N$. We get that $\sum_{j\in\mathbb{Z}}[s(i+1,j)-s(i,j)]\in[-M/N,M/N]$. As N may be chosen arbitrarily large, then $\sum_j s(i+1,j)=\sum_j s(i,j)$. From that, the sum of every rank $\sum_{j\in\mathbb{Z}} s(i,j)$ is the same.

Let $i\in\mathbb{Z}$, we choose an interval $[a,b]$, $a,b\in\mathbb{Z}$, such that $s(i,j)=0$ for all $j\notin[a,b]$. For every $j\in[a,b]$, we define the function $f(\ell)=\delta_{\ell,j}$. Let $g=Sf$, then $\Delta g(k)=s(k+1,j)-s(k,j)$. Since $\delta(f)=1$, then $\delta(g)\le M$, i.e. $|s(k+1,j)-s(k,j)|\le M$ for all k and j. Let us assume that $s(i,j)\neq 0$, then $i-2j\in[\sigma,\sigma']$. We set $N=\sigma'-\sigma$, then $s(i+N+1,j)=0$. From the telescoping sum, $\sum_{k=0}^{N}[s(i+k+1,j)-s(i+k,j)]=s(i+N+1,j)-s(i,j)$, we obtain that $|s(i,j)|\le M(N+1)$. The entries $s(i,j)$ are uniformly bounded.

In the second part of the proof, we assume that $\sum_{j\in\mathbb{Z}} s(i,j)=C$ for all i and $|s(i,j)|\le M$ for all i and j. Let $f\in\ell_0(\mathbb{Z})$ such that $\delta(f)=||\Delta f||_\infty\le 1$, let $g=Sf$ and let $J=\{j : i-2j \text{ or } i+1-2j\in[\sigma,\sigma']\}$, then $\Delta g(i)=\sum_{j\in J}[s(i+1,j)-s(i,j)]f(j)$. Let k be the minimal element of J, since the sum of every rank is the same, $\Delta g(i)=\sum_{j\in J}[s(i+1,j)-s(i,j)][f(j)-f(k)]$ and $|\Delta g(i)|\le 2M\sum_{j\in J}(j-k)\le 2M(\sigma'-\sigma)^2$. The subdivision operator S is bounded with respect to the difference norm δ. □

The following lemma is Lemma 1 of [5].

Lemma 2. *Let S be a subdivision matrix with support $[\sigma,\sigma']$ and let $S^n=(s_n(i,j) : i,j\in\mathbb{Z})$ be the n-th power of S with $n>0$. If $s_n(i,j)\neq 0$, then $i-2^n j\in(2^n-1)[\sigma,\sigma']$.*

Proof. We proceed by induction on n. For $n=1$, it is tautological. Let m be a positive integer. Let us assume by induction, that the lemma is true for $n=m$, we will show that the lemma is true for $n=m+1$. Let $s_{m+1}(i,j)$ be a nonzero entry of S^{m+1}, since $s_{m+1}(i,j)=\sum_{k\in\mathbb{Z}} s_m(i,k)s(k,j)$, there exists k such $s_m(i,k)\neq 0$ and $s(k,j)\neq 0$. By hypothesis, $i-2^m k\in(2^m-1)[\sigma,\sigma']$ and $k-2j\in[\sigma,\sigma']$. We obtain

$$i-2^{m+1}j=(i-2^m k)+2^m(k-2j)\in(2^m-1)[\sigma,\sigma']+2^m[\sigma,\sigma']=(2^{m+1}-1)[\sigma,\sigma'].$$

the lemma is true for $n=m+1$. □

Theorem 3. *Let $\mathscr{S}$ be a subdivision scheme whose subdivision operator S on $\ell_0(\mathbb{Z})$ is affine, we assume that there exist $n \in \mathbb{N}$ such that $||S^n|| < 1$ for a norm $||\cdot||$ equivalent to the difference norm δ, then $\mathscr{S}$ is C^0 convergent. Moreover, on any interval, every limit function ϕ of $\mathscr{S}$ is Hölder continuous*

$$|\phi(x) - \phi(y)| \leq C|x-y|^\alpha$$

with $\alpha = -\ln||S^n||^{1/n}/\ln 2$. Finally if not only $||S^n|| < 1$, but $||S^n|| < 1/2^n$, then any limit function of $\mathscr{S}$ is constant.

Proof. The norm $||\cdot||$ is equivalent to the difference norm, there exist $\varepsilon > 0$ and M such that $\varepsilon\delta(f) \leq ||f|| \leq M\delta(f)$ for all $f \in \ell_0(\mathbb{Z})$. Let $m \in \mathbb{N}$ be such that $||S^m|| < 1$. We assume that $[\sigma, \sigma']$ is a support for $\mathscr{S}$. We consider a sequence of refinements $f_n : \mathbb{Z} \to \mathbb{R}$ of $\mathscr{S}$ and the corresponding sequence of piecewise linear functions $\phi_n : \mathbb{R} \to \mathbb{R}$ with nodes $i/2^n$, $i \in \mathbb{Z}$ and with values $\phi_n(i/2^n) = f_n(i)$. If $a < b$ are two integers of $\mathbb{Z}$, let $||\cdot||_\infty$ denote the uniform norm on $C[a,b]$. We will show that ϕ_n converge to a function ϕ in the uniform norm.

The values of ϕ_n on $[a,b]$ are uniquely determined by $f_n(i), i \in [a2^n, b2^n]$. Since

$$(\forall i \in \mathbb{Z}) f_n(i) = \sum_{j\in\mathbb{Z}} s_n(i,j) f_0(j),$$

it follows from Lemma 2 that the restriction of ϕ_n to $[a,b]$ is uniquely determined by the values of $f_0(j)$, $j \in J$ where $J = [c,d]$, $c = \inf_n\{(a - 1/2^n)\sigma'\}$ and $d = \sup_n\{(b - 1/2^n)\sigma\}$. So there is no loss of generality by assuming that $f_0(j) = 0, j \notin J$.

We compare both sequences $||\phi_{n+1} - \phi_n||_\infty$ and $||f_n||$ (the norm of f_n in $\ell_0(\mathbb{Z})$). Since the maximum difference on $[a,b]$ between ϕ_{n+1} and ϕ_n is attained at a point on the $(n+1)$th mesh, then

$$||\phi_{n+1} - \phi_n||_\infty \leq \max\{M_n, M'_n\}, \tag{2}$$

where

$$\begin{cases} M_n = \max\{|f_{n+1}(2i) - f_n(i)| : i \in \mathbb{Z}\} \\ M'_n = \max\{|f_{n+1}(2i+1) - (f_n(i) + f_n(i+1))/2| : i \in \mathbb{Z}\}. \end{cases} \tag{3}$$

As S is affine, we obtain

$$f_{n+1}(2i) - f_n(i) = \sum_{\sigma \leq 2i-2j \leq \sigma'} s(2i,j)(f_n(j) - f_n(i)).$$

From the telescoping series $\sum_{k=i}^{i+N-1} \Delta f(k) = f(i+N) - f(i)$, we obtain the inequality $|f(i \pm N) - f(i)| \leq N\delta(f)$. Since $|f_n(j) - f_n(i)| \leq |j-i|\delta(f_n) \leq |j-i| \cdot ||f_n||/\varepsilon$, we obtain $|f_{n+1}(2i) - f_n(i)| \leq \beta||f_n||$ where

$$\beta = \sup_{i\in\mathbb{Z}} \sum_{\sigma \leq 2i-2j \leq \sigma'} |s(2i,j)(j-i)|/\varepsilon.$$

From (3), we get $M_n \leq \beta||f_n||$.

The number $2f_{n+1}(2i+1) - f_n(i) - f_n(i+1)$ is the sum of the four terms $f_{n+1}(2i) - f_n(i)$, $f_{n+1}(2i+2) - f_n(i+1)$, $f_{n+1}(2i+1) - f_{n+1}(2i)$ and $f_{n+1}(2i+1) - f_{n+1}(2i+2)$.

From (3), it follows that $M'_n \leq M_n + \delta(f_{n+1})$. From (2) and from the inequality $\delta(f_{n+1}) \leq M||f_{n+1}||$, we obtain

$$||\phi_{n+1} - \phi_n||_\infty \leq \beta||f_n|| + M||f_{n+1}||. \tag{4}$$

We set $\lambda = ||S^m||^{1/m}$. By hypothesis, $\lambda < 1$. By definition of $||S^m||$, for every function f, $||S^m f|| \leq \lambda^m||f||$. From that $||f_{n+m}|| \leq \lambda^m||f_n||$, $n = 0, 1, \ldots$. We obtain the inequalities $||f_{mq+r}|| \leq \lambda^{mq}||f_r||$ for $r = 0, 1 \ldots, m-1$, $q = 0, 1, 2, \ldots$, and

$$||f_n|| \leq L\lambda^n \tag{5}$$

where $L = \max\{||f_r||/\lambda^r : r = 0, 1, \ldots, m-1\}$.

From (4) and (5), we get

$$\sum_{n=0}^{\infty} ||\phi_{n+1} - \phi_n||_\infty \leq L(\beta + M\lambda)/(1-\lambda).$$

From Weierstrass criterion, the sequence ϕ_n converges uniformly on $[a,b]$ to a function ϕ. The interval $[a,b]$ may be arbitrary large, the scheme $\mathscr{S}$ is C^0 convergent.

Let us investigate the Hölder continuity. Let $x, y \in \mathbb{R}$ be such that $0 < y - x \leq 1$, there exists $\ell \in \mathbb{N} \cup 0$ such that $1/2^{\ell+1} < y - x \leq 1/2^\ell$. Then

$$|\phi(x) - \phi(y)| \leq |\phi(x) - \phi_\ell(x)| + |\phi_\ell(x) - \phi_\ell(y)| + |\phi_\ell - \phi(y)|. \tag{6}$$

Every function ϕ_ℓ is a Lipschitz function, the Lipschitz constant being the largest slope between two consecutive nodes:

$$|\phi_\ell(x) - \phi_\ell(y)| \leq 2^\ell \delta(f_\ell)(y - x) \leq M||f_\ell||. \tag{7}$$

From the fact that $\phi - \phi_\ell = \sum_{n=\ell}^{\infty}(\phi_{n+1} - \phi_n)$ and from (6) and (7), it follows that

$$|\phi(x) - \phi(y)| \leq 2(\beta + M) \sum_{n=\ell}^{\infty} ||f_n||. \tag{8}$$

From (5) and (8), we obtain that

$$|\phi(x) - \phi(y)| \leq 2(\beta + M)L\lambda^\ell/(1-\lambda)$$

Let α be the exponent for which $1/2^\alpha = \lambda$. Since $1/2^\ell \leq 2(y-x)$, we get $\lambda^\ell \leq [2(y-x)]^\alpha$ and $|\phi(x) - \phi(y)| \leq C|x-y|^\alpha$ where $C = 2(\beta + M)L/(\lambda - \lambda^2)$.

Finally if $||S^n|| < 1/2^n$, then the number $-\ln||S^n||^{1/n}/\ln 2$ is > 1. Let ϕ be any limit function $\mathscr{S}$, then $\phi(x+h) - \phi(x) = o(h)$ as $h \to 0$, this means that $\phi'(x) = 0$ for all x and ϕ is a constant function. □

Theorem 3 is an extension of Lemma 3.1 of Dyn, Gregory and Levin [6].

3 Computation of $\delta(S^n)$

We provide two ways for computing the difference norm of S^n: the primal formula (9) and the dual formula (10).

Theorem 4. *Let S be an affine subdivision matrix with support $[\sigma,\sigma']$, for every $i\in\mathbb{Z}$, we set $[a_i,b_i]=([i,i+1]+(2^n-1)[-\sigma',-\sigma])/2^n\cap\mathbb{Z}$, then*

$$\delta(S^n)=\sup_{i\in\mathbb{Z}}\{\sum_{j=a_i}^{b_i}[s_n(i+1,j)-s_n(i,j)]f(j):f\in\mathscr{E}_i\} \tag{9}$$

where $\mathscr{E}_i=\{f:[a_i,b_i]\to\mathbb{R}:f(a_i)=0,|\Delta f|=1\}$.

Proof. By definition, $\delta(S^n)=\sup\{\delta(S^nf):\delta(f)\le 1\}$, i.e.

$$\delta(S^n)=\sup_{\delta(f)\le 1}\sup_{i\in\mathbb{Z}}\{\sum_{j\in\mathbb{Z}}[s_n(i+1,j)-s_n(i,j)]f(j)\}.$$

By changing the order of the upper bounds, we get $\delta(S^n)=\sup_i m(i)$ where

$$m(i)=\sup\{\sum_{j\in\mathbb{Z}}[s_n(i+1,j)-s_n(i,j)]f(j):f\in\ell_0(\mathbb{Z}),|\Delta f|\le 1\}.$$

Let $i\in\mathbb{Z}$. If $s_n(i+1,j)-s_n(i,j)\neq 0$, then by Lemma 2, $2^nj\in[i,i+1]+(2^n-1)[-\sigma',-\sigma]$. It follows that

$$m(i)=\sup\{\sum_{j=a_i}^{b_i}[s_n(i+1,j)-s_n(i,j)][f(j)-f(a_i)]:\Delta f\le 1\}$$

since S is affine. The number $m(i)$ is the the least upper bound of the values of a linear function on a compact convex set $K_i=\{f:[a_i,b_i]\to\mathbb{R}:f(a_i)=0,|\Delta f|\le 1\}$. After identifying the extreme points of K_i, we obtain (9). □

Remark 5. The cardinality of $\mathscr{E}_i$ is equal to $2^{\sigma'-\sigma}$. If S is periodic, the computation of $\delta(S^n)$ involves a finite number of arithmetical operations.

Before getting the dual formula of $\delta(S^n)$, we investigate the dual space of $\ell_0(\mathbb{Z})$.

Lemma 6. *Let the seminormed space $E=\{f:\mathbb{Z}\to\mathbb{R}:\lim_{i\to\pm\infty}\Delta f(i)=0\}$ with its seminorm $\delta(f)=\max\{|\Delta f(i)|:i\in\mathbb{Z}|\}$, then $\ell_0(\mathbb{Z})$ is dense in E.*

Proof. Let $\psi:\mathbb{R}\to[0,1]$ be the function such that $\psi(x)=1$ if $|x|\le 1$, $\psi(x)=1-|x|$ if $1<|x|<2$ and $\psi(x)=0$ if $|x|\ge 2$. The function ψ is a Lipschitz function: $|\psi(x)-\psi(y)|\le|x-y|$ for all $x,y\in\mathbb{R}$. Let $f\in E$ and let $\varepsilon>0$, there exists an integer M such that $|\Delta f(i)|<\varepsilon$ whenever $|i|\ge M$. We set $a=\max\{|f(i)|:i\in[-M,M]\}$.

Let $N>M$, we set $f_N(i)=\psi(i/N)f(i)$ for all $i\in\mathbb{Z}$. Then

$$\Delta f_N(i)-\Delta f(i)=[\psi((i+1)/N)-\psi(i/N)]f(i)+[\psi((i+1)/N)-1]\Delta f(i).$$

If $|i| > 2N$, then $|[\psi((i+1)/N) - \psi(i/N)]f(i)| = 0$. If $|i| \leq 2N$, then $|[\psi((i+1)/N) - \psi(i/N)]f(i)| \leq |f(i)|/N$ (from the Lipschitz property of ψ) and $|f(i)| \leq a + 2N\varepsilon$ (if $i > M$, then $f(i) - f(M) = \sum_{j=M}^{i-1} \Delta f(j)$ and $|f(i)| \leq |f(M)| + 2N\varepsilon$; a similar inequality holds for $i < -M$; if $i \in [-M, M]$, the inequality $|f(i)| \leq a + 2N\varepsilon$ is obvious). Thus for all i, $|[\psi((i+1)/N) - \psi(i/N)]f(i)| \leq (a + 2N\varepsilon)/N$.

If $|i| \geq M$, then $|[\psi((i+1)/N) - 1]\Delta f(i)| < \varepsilon$ (for every x, $|\psi(x) - 1| \leq 1$). If $|i| < M$, then $[\psi((i+1)/N) - 1]\Delta f(i) = 0$.

If follows that

$$|\Delta f_N(i) - \Delta f(i)| \leq 3\varepsilon + a/N.$$

We may choose N so large that $a/N < \varepsilon$ and in that case, $|\Delta f_N(i) - \Delta f(i)| \leq 4\varepsilon$ for all $i \in \mathbb{Z}$. Obviously $f_N \in \ell_0(\mathbb{Z})$. $\ell_0(\mathbb{Z})$ is dense in E. □

Theorem 7. *Let $\lambda : \mathbb{Z} \to \mathbb{R}$, we consider the linear functional $L : \ell_0(\mathbb{Z}) \to \mathbb{R}$, $L(f) = \sum_{i\in\mathbb{Z}} \lambda(i) f(i)$. The functional L is bounded on the normed space $\ell_0(\mathbb{Z})$ with the difference norm δ if and only if there exists a function $\Lambda : \mathbb{Z} \to \mathbb{R}$ such that $\lambda = \Delta\Lambda$ and $\sum_{i\in\mathbb{Z}} |\Lambda(i)| < \infty$. Moreover in that case, $\sum_{i\in\mathbb{Z}} |\Lambda(i)| = ||L||$.*

Proof. Sufficiency. Let us assume that there exists a function $\Lambda : \mathbb{Z} \to \mathbb{R}$ such that $\lambda = \Delta\Lambda$ and $\sum_{i\in\mathbb{Z}} |\Lambda(i)| < \infty$. Let $f \in \ell_0(\mathbb{Z})$, by summation by parts

$$L(f) = \sum_{i\in\mathbb{Z}} [\Delta\Lambda(i)] f(i) = -\sum_{i\in\mathbb{Z}} \Lambda(i+1)[\Delta f(i)].$$

If $|\Delta f| \leq 1$, then $|L(f)| \leq \sum_{i\in\mathbb{Z}} |\Lambda(i)|$.

Necessity. Let us assume that L is a linear functional on $\ell_0(\mathbb{Z})$ such that $|L(f)| \leq M||\Delta f||_\infty$ for all f. Let $E = \{f : \mathbb{Z} \to \mathbb{R} : \lim_{i\to\infty} \Delta f(i) = 0\}$ and let $||\Delta f||_\infty$ be the seminorm on E. By Lemma 6, L can be extended on the whole E by continuity and $|L(f)| \leq M||\Delta f||_\infty$ for all $f \in E$. For every $j \in \mathbb{Z}$, we define the function $f_j : \mathbb{Z} \to \mathbb{R}$, $f_j(i) = 1$ if $i \geq j$ and $f_j(i) = 0$ otherwise; f_j belongs to E. We define the function Λ, $\Lambda(j) = -L(f_j)$. The function $f_j - f_{j+1}$ takes the value 1 only at j, otherwise it is 0. From that, $\Delta\Lambda(j) = L(f_j - f_{j+1}) = \lambda(j)$.

Let N be an arbitrary integer, for $j \in [-N, N]$, we define s_j as -1 if $\Lambda(j) \geq 0$ and 1 otherwise. Let $f = \sum_{j=-N}^{N} s_j f_j$. Since $\Delta f(i) = \sum_{j=-N}^{N} s_j \delta_{i,j-1}$, the values taken by Δf are 0 or ± 1. Thus $L(f) = \sum_{j=-N}^{N} |\Lambda(j)| \leq M$. It follows that $\sum_{j\in\mathbb{Z}} |\Lambda(j)| \leq ||L||$ where $||L|| = \sup\{L(f) : f \in \ell_0(\mathbb{Z}), |\Delta f| \leq 1\}$.

Now let $\varepsilon > 0$, then there exists $f \in \ell_0(\mathbb{Z})$ such that $|\Delta f| \leq 1$ and $L(f) > ||L|| - \varepsilon$. Since $f = \sum_j f(j)(f_j - f_{j+1})$, then

$$L(f) = \sum_j f(j)[\Lambda(j+1) - \Lambda(j)] = -\sum_j \Lambda(j+1)[f(j+1) - f(j)].$$

It follows that

$$||L|| - \varepsilon \leq L(f) \leq \sum_{i\in\mathbb{Z}} |\Lambda(i)|.$$

As ε may be arbitrary small, we obtain $||L|| \leq \sum_{i\in\mathbb{Z}} |\Lambda(i)|$. The conclusion is reached. □

Remark 8. If $L(f)=\sum_{i\in\mathbb{Z}}\lambda(i)f(i)$ is a bounded linear functional on $(\ell_0(\mathbb{Z}),\delta)$, then $\sum_i|\lambda(i)|<\infty$ and $\sum_i\lambda(i)=0$. This follows from the fact that $\lambda=\Delta\Lambda$ with $\Lambda\in\ell^1$. More precisely, $\lambda(i)=\Lambda(i+1)-\Lambda(i)$ with $\sum_i|\Lambda(i)|<\infty$; thus $\sum_i|\lambda(i)|\leq 2\sum_i|\Lambda(i)|$ and $\sum_i\lambda(i)=\sum_i\Delta\Lambda(i)=0$.

There is another way for computing $\delta(S^n)$, by duality. Let S^* be the adjoint of S considered as an operator on the normed space $(\ell_0(\mathbb{Z}),\delta)$. It means that for all $L\in\ell_0(\mathbb{Z})^*$, for $f\in\ell_0(\mathbb{Z})$, $S^*L(f)=L(Sf)$. We recall a theorem that is proved by Akilov and Kantorovic [8, p.305]. *If $U:E\to E$ is a continuous linear mapping on the normed space E, then U^* is a continuous linear mapping on E^* and $||U^*||=||U||$.* From that, it follows that $\delta(S^n)=\delta(S^{*n})$ for every n.

Theorem 9. *Let S be an affine subdivision matrix and let T be the matrix whose entries are*

$$t(i,j)=\sum_{k=-\infty}^{j}[s(i,k)-s(i+1,k)],$$

then for all $n\in\mathbb{N}$,

$$\delta(S^n)=||T^n||_\infty. \tag{10}$$

Proof. By definition,

$$\delta(S^{*n})=\sup\{||S^{*n}L||:L\in\ell_0(\mathbb{Z})^*,||L||\leq 1\}.$$

It is an optimization problem, the maximization of a convex function on the unit ball B of $\ell_0(\mathbb{Z})^*$. Let $j\in\mathbb{Z}$, we define the linear functional L_j: $L_j(f)=\Delta f(j)$. Then the set $\mathscr{E}$ of extreme points of B is precisely $\{\pm L_j:j\in\mathbb{Z}\}$. The convex hull of points of $\mathscr{E}$ is dense in B. Consequently $\delta(S^{*n})=\sup\{||S^{*n}L_j||:j\in\mathbb{Z}\}$. Let $i\in\mathbb{Z}$, then $S^{*n}L_i(f)=\sum_j\lambda(j)f(j)$ where $\lambda(j)=s(i+1,j)-s(i,j)$. According to Theorem 7, the function $\Lambda(j)=t_n(i,j)=\sum_{k\leq j}[s(i+1,k)-s(i,k)]$ is $\ell^1(\mathbb{Z})$ and $||S^{*n}L_i||=\sum_j|t(i,j)|$. We get the following

$$\delta(S^{*n})=\sup_i\{\sum_j|t_n(i,j)|\},$$

which is (10). □

Corollary 10. *If there exists an n such that $||T^n||_\infty<1$, then the corresponding subdivision scheme is C^0.*

Buhmann and Micchelli [1, Theorem 3.3] proved the previous result for periodic subdivision schemes. See also Theorem 4 of Charina, Conti and Sauer [3] where they investigate the C^0 convergence of uniform multivariate vector subdivision schemes.

4 Contraction of the Modulus of Continuity

Let $f:\mathbb{Z}\to\mathbb{R}$, $h\in\mathbb{N}$, the *modulus of continuity* of f is

$$\omega_h(f)=\sup\{|f(i)-f(i')|:|i-i'|\leq h\}.$$

For every h, on $\ell_0(\mathbb{Z})$, ω_h is a norm equivalent to the difference norm (since $\delta(f)\leq\omega_h(f)\leq h\delta(f)$). Let S be a subdivision matrix, $s_n(i,j)$ is the (i,j)-entry of S^n.

Theorem 11. *Let S be an affine subdivision matrix whose entries are uniformly bounded and whose support is $[\sigma,\sigma']$, let $h \geq \sigma' - \sigma - 1$ and let the norm $\omega_h(f)$. Then for $n = 1,2,3,\ldots$*

$$\omega_h(S^n) = \sup\{\sum_{j\in\mathbb{Z}} |s_n(i,j) - s_n(i',j)|/2 : i,i' \in \mathbb{Z}, |i-i'| \leq h\}$$

provides the norm of S^n with respect to the norm ω_h on $\ell_0(\mathbb{Z})$.

Proof. We set $\mu_n = \sup\{\sum_{j\in\mathbb{Z}} |s_n(i,j) - s_n(i',j)|/2 : i,i' \in \mathbb{Z}, |i-i'| < h\}$.

First step of the proof: $\omega_h(S^n) \leq \mu_n$. Let $f : \mathbb{Z} \to \mathbb{R}$ such that $\omega_h(f) \leq 1$, we will show $\omega_h(g) \leq \mu_n$ where $g = S^n f$. Since the subdivision scheme is affine, $\sum_{j\in\mathbb{Z}} s_n(i,j) = 1, i \in \mathbb{Z}$. We deduce that for any $c \in \mathbb{R}$,

$$g(i) - c = \sum_{j\in\mathbb{Z}} s_n(i,j)(f(j) - c), i \in \mathbb{Z}. \tag{11}$$

We consider two integers $i_1, i_2 \in \mathbb{Z}$ with $|i_1 - i_2| \leq h$. We may assume that $i_1 \leq i_2$ and we define the sets $J_1 = \{j : -\sigma'(2^n-1) + i_1 \leq 2^n j \leq -\sigma(2^n-1) + i_1\}$, $J_2 = \{j : -\sigma'(2^n-1) + i_2 \leq 2^n j \leq -\sigma(2^n-1) + i_2\}$ and $J = \{j : -\sigma'(2^n-1) + i_1 \leq 2^n j \leq -\sigma(2^n-1) + i_2\}$. From the fact that $J_1 \subset J$ and $J_2 \subset J$ and from Lemma 2, we obtain that $s_n(i_1,j) = s_n(i_2,j) = 0$ for any $j \notin J$.

Using equality (11) for i_1 and i_2 we get:

$g(i) - c = \sum_{j\in J} s_n(i,j)(f(j) - c)$ for $i = i_1$ and i_2.

By substraction, for any $c \in \mathbb{R}$ we obtain:

$g(i_1) - g(i_2) = \sum_{j\in J} [s_n(i_1,j) - s_n(i_2,j)](f(j) - c)$, from which we deduce that:

$$\forall c \in \mathbb{R}, |g(i_1) - g(i_2)| \leq 2\mu_n \max\{|f(j) - c| : j \in J\}.$$

Now let $M = \max\{f(j) : j \in J\}$ and $m = \min\{f(j) : j \in J\}$, we choose $c = (M+m)/2$. We already know that for any $j \in J$, $|f_n(j) - c| \leq (M-m)/2$. We get $|g(i_1) - g(i_2)| \leq \mu_n(M-m)$.

The diameter of J is $\delta(J) = [i_2 - i_1 + (\sigma' - \sigma)(2^n-1)]/2^n$. By hypothesis, $\sigma' - \sigma \leq h+1$, we obtain that $\delta(J) \leq h + (2^n-1)/2^n$. Since $\delta(J)$ is an integer, we get $\delta(J) \leq h$ and $M - m \leq \omega_h(f) \leq 1$. It follows that $|g(i_1) - g(i_2)| \leq \mu_n$ and $\omega_h(S^n f) \leq \mu_n \omega_h(f)$ for all $f \in \ell_0(\mathbb{Z})$.

Second step of the proof: $\omega_h(S^n) \geq \mu_n$. Let $\varepsilon \in (0, \mu_n)$, then there exist $i_1, i_2 \in \mathbb{Z}$ such that $|i_1 - i_2| \leq h$ and

$$\sum_{j\in\mathbb{Z}} |s_n(i_1,j) - s_n(i_2,j)|/2 > \mu_n - \varepsilon.$$

We set $f(j) = 1$ if $[s_n(i_1,j) - s_n(i_2,j)] > 0$, $f(j) = 0$ if $s_n(i_1,j) = s_n(i_2,j)$ and $f(j) = -1$ otherwise. Since $\sum_{j\in\mathbb{Z}} |s_n(i_1,j) - s_n(i_2,j)| > 0$ and $\sum_{j\in\mathbb{Z}} [s_n(i_1,j) - s_n(i_2,j)] = 0$, there exist j_1 and j_2 such that $f(j_1) = 1$ and $f(j_2) = -1$. Let us show that that $|j_2 - j_1| \leq h$. Since $s_n(i_1,j_1) \neq s_n(i_2,j_1)$, there exists $i \in \{i_1,i_2\}$ such that $s_n(i,j_1) \neq 0$.

Similarly, there exists $i' \in \{i_1, i_2\}$ such that $s_n(i', j_2) \neq 0$. From Lemma 2, $i - 2^n j_1 \in (2^n - 1)[\sigma, \sigma']$ and $i' - 2^n j_2 \in (2^n - 1)[\sigma, \sigma']$. After substraction, we obtain $2^n |j_1 - j_2| \leq |i - i'| + (2^n - 1)(\sigma, -\sigma)) \leq 2^n h + 2^n - 1$. It follows that $|j_1 - j_2| \leq h$. This shows that $\omega_h(f) = 2$.

Moreover if we set $g = S^n f$, then $g(i_1) - g(i_2) = \sum_{j \in J} \sum_{j \in \mathbb{Z}} |s_n(i_1, j) - s_n(i_2, j)| > 2(\mu_n - \varepsilon)$, this means that $\omega_h(g) \geq 2(\mu_n - \varepsilon)$. As ε may be arbitrarily small, we get $\omega_h(S^n) \geq \mu_n$. □

We are ready for our second criterion of convergence.

Corollary 12. *Let $\mathscr{S}$ be a subdivision scheme whose subdivision matrix S is affine and has $[\sigma, \sigma']$ as support. We assume that there exist $n > 0$ and $\alpha < 2$ such that $\sum_{j \in \mathbb{Z}} |s_n(i, j) - s_n(i', j)| \leq \alpha$ whenever $|i - i'| < \sigma' - \sigma$. Then $\mathscr{S}$ is C^0.*

Proof. Let $N = \sigma' - \sigma$, from Theorem 11, there exists an integer n such that $\omega_N(S^n) < 1$. If we apply Theorem 3 with the norm $||f|| = \omega_N(f)$, then $\mathscr{S}$ is C^0. □

5 Contraction of Joint Norms

In this section, we limit ourselves to affine periodic subdivision schemes. For these schemes, we propose a third criterion that originates in [4]. This criterion involves joint norms and is linked to the second criterion by duality. We begin by proving some lemmas about periodic subdivision schemes.

Lemma 13. *Let S be a periodic subdivision matrix S of period p and let $s_n(i, j)$ be the entries of the matrix S^n for $n > 0$, then all $i, j \in \mathbb{Z}$ $s_n(i + p2^n, j + p) = s_n(i, j)$.*

Proof. We proceed by induction. The entry $s_{n+1}(i + p2^{n+1}, j + p)$ is equal to $\sum_{k \in \mathbb{Z}} s(i + p2^{n+1}, k) s_n(k, j + p) = \sum_{k \in \mathbb{Z}} s(i, k - p2^n) s_n(k - p2^n, j)$ by periodicity of S and by induction, and this is equal to $\sum_{k \in \mathbb{Z}} s(i, k) s_n(k, j) = s_{n+1}(i, j)$. □

Lemma 14. *Let S be a subdivision matrix of support $[\sigma, \sigma']$. Let $p > 0$, $n > 0$ and $S^n = (s_n(i, j) : i, j \in \mathbb{Z})$, if $i \in [-\sigma' + 1, -\sigma + p2^n - 1]$ and $s_n(i, j) \neq 0$, then $j \in [-\sigma' + 1, -\sigma + p - 1]$.*

Proof. Let i and j be such that $i \in [-\sigma' + 1, -\sigma + p2^n - 1]$ and $s_n(i, j) \neq 0$. From Lemma 2, $i - 2^n j \in (2^n - 1)[\sigma, \sigma']$ and $2^n j \in i + (2^n - 1)[-\sigma', -\sigma]$. We get that $2^n j \in [-2^n \sigma' + 1, -2^n \sigma + p2^n - 1]$. It follows that $j \in [-\sigma' + 1, -\sigma + p - 1]$. □

Lemma 15. *Let S be a periodic subdivision matrix of support $[\sigma, \sigma']$ and of period p. Let $N = \sigma' - \sigma + p - 1$, define for $n = 1, 2, \ldots$, the nth family of $N \times N$ matrices*

$$A_r^{(n)} = (s_n(i + pr, j) : i, j \in [-\sigma' + 1, -\sigma + p - 1]), \quad r = 0, 1, \ldots, 2^n - 1$$

and set $A_0 = A_0^{(1)}$, $A_1 = A_1^{(1)}$. Then $A_r^{(n)} = A_{b_1} A_{b_2} \ldots A_{b_n}$ where $\sum_{k=1}^n b_k 2^{k-1}$ is the binary expansion of r.

Proof. We proceed by induction on n. For $n=1$, the statement is trivial. Let us assume that the lemma is true $n=m$ where m is a given value. Let $n=m+1$ and $r\in[0,2^{m+1}-1]$, we use the binary expansion of r, $\sum_{k=1}^{m+1} b_k 2^{k-1}$. Let $i,j\in[-\sigma'+1,-\sigma+p-1]$, then

$$s_{m+1}(i+pr,j)=\sum_{k\in\mathbb{Z}} s_m(i+pr,k)s(k,j).$$

We distinguish two cases.

Case 1. $b_{m+1}=0$. By replacing in Lemma 14, n by m, i by $i+pr$ and j by k, we obtain

$$s_{m+1}(i+pr,j)=\sum_{k\in[-\sigma'+1,-\sigma+p+1]} s_m(i+pr,k)s(k,j). \tag{12}$$

and $A_r^{(m+1)}=A_{b_1}A_{b_2}\ldots A_{b_m}A_0$.

Case 2. $b_{m+1}=1$. From Lemma 13,

$$\begin{aligned} s_{m+1}(i+pr,j)&=\sum_{k\in\mathbb{Z}} s_m(i+pr-p2^m,k-p)s(k,j),\\ &=\sum_{k\in\mathbb{Z}} s_m(i+pr-p2^m,k)s(k+p,j). \end{aligned}$$

By replacing in Lemma 14, n by m, i by $i+pr$ and j by k, we obtain

$$s_{m+1}(i+pr,j)=\sum_{k\in[-\sigma'+1,-\sigma+p-1]} s_m(i+pr-p2^m,k)s(k+p,j). \tag{13}$$

and $A_r^{(m+1)}=A_{b_1}A_{b_2}\ldots A_{b_m}A_1$.

In both cases, we obtain the desired conclusion. □

Lemma 16. *Let $n>1$ and let (a_{ij}) be an $n\times n$ matrix, then*

$$\max\{\sum_i|\sum_j x_i a_{ij}|:\sum_i x_i=0,\sum_i|x_i|\le 1\}=\max\{\sum_j|a_{ij}-a_{i'j}|/2:i<i'\}.$$

Proof. Let K be the compact convex subset of $\mathbb{R}^n$, $\{(x_1,x_2,\ldots,x_n):\sum_i x_i=0,\sum_i|x_i|\le 1\}$, the extreme points of K are

$$\{\pm(x_1,x_2,\ldots,x_n):\exists i'<i'' \text{ such that } x_{i'}=1/2,x_{i''}=-1/2,x_i=0 \text{ for } i\neq i',i''\}.$$

The maximum of the convex function $f(x_1,x_2,\ldots x_n)=\sum_i|\sum_j x_i a_{ij}|$ is reached at an extreme point, this maximum is $\max\{\sum_j|a_{i'j}-a_{i''j}|/2:k<\ell\}$. □

Let $\mathscr{A}$ be a finite set of square matrices of order m, the *n-th joint norm of $\mathscr{A}$* is defined to be

$$J_n(\mathscr{A},||\cdot||)=\max\{||A_1A_2\ldots A_n||:A_i\in\mathscr{A},i=1,2,\ldots,n\}$$

where $||\cdot||$ is a matrix norm acting on $m\times m$ matrices. This notion of joint norms originates from Rota and Strang [9].

Theorem 17. *Let S be an affine subdivision matrix of support $[\sigma,\sigma']$ which is periodic of period p. Let $N=\sigma'-\sigma+p-1$, define two $N\times N$ matrices*

$$A_r=(s(i+pr,j):i,j\in[-\sigma'+1,-\sigma+p-1]),\quad r=0,1.$$

Define E to be the $(N-1)$-dimensional subspace of $\mathbb{R}^N$ orthogonal to $(1,\ldots,1)$. We choose the 1-norm of $\mathbb{R}^N$ restricted to E. Let the norm ω_N on $\ell_0(\mathbb{Z})$. Then for $n=1,2,3,\ldots$,

$$J_n(A_0^T|E,A_1^T|E,||\cdot||_1)=\omega_N(S^n). \tag{14}$$

Proof. Since S is affine, the column vector with N equal components is a common eigenvector of A_0,A_1. From that, A_0^T and A_1^T map E into itself. If B_r is the restriction of A_r^T to E, we may compute the joint norms $J_n(B_0,B_1)$ from any norm of E. We choose the 1-norm of $\mathbb{R}^N$ restricted to E: $||x||_1=\sum_{i=1}^N|x_i|$ where $x=(x_1,\ldots,x_N)^T\in E$. We consider the norm $\omega_N(f)$ on $\ell_0(\mathbb{Z})$. We will prove that for $n=1,2,3,\ldots$,

$$\max\{||B_{b_1}B_{b_2}\ldots B_{b_n}||_1:b_1=0,1,b_2=0,1,,\ldots,b_n=0,1,\}=\omega_N(S^n). \tag{15}$$

We set $J=[-\sigma'+1,-\sigma+p-1]$. Let $n>0$ and let r be an integer $\in[0,2^n-1]$, we use the binary expansion of r: $\sum_{k=1}^n b_k2^{k-1}=r$. From Lemma 15, the matrix $A_{b_1}A_{b_2}\ldots A_{b_n}$ is equal to $(s_n(i+r,j:i,j\in J)=(c_{ij}:i,j=1,\ldots,N)$. From Lemmas 16 and 14, we obtain that

$$\begin{aligned}
||B_{b_n}B_{b_{n-1}}\ldots B_{b_1}||_1&=\max\{\sum_i|\sum_j x_ic_{ij}|:\sum_i x_i=0,\sum_i|x_i|\le 1\},\\
&=\max\{\sum_j|c_{ij}-c_{i'j}|/2:i<i'\},\\
&=\max\{\sum_{j\in J}|s_n(i+r,j)-s_n(i'+r,j)|/2:i,i'\in J,i<i'\},\\
&=\max\{\sum_{j\in\mathbb{Z}}|s_n(i+r,j)-s_n(i'+r,j)|/2:i,i'\in J\}.
\end{aligned}$$

From Lemma 13 and Theorem 11, we obtain

$$\begin{aligned}
&\max\{\sum_{j\in\mathbb{Z}}|s_n(i+r,j)-s_n(i'+r,j)|/2:i,i'\in J,r\in[0,2^n-1]\},\\
=\max\{\sum_{j\in\mathbb{Z}}&|s_n(i+r,j)-s_n(i'+r,j)|/2:i,i'\in\mathbb{Z},|i'-i|\le N\}=\omega_N(S^n).
\end{aligned}$$

From that, (15) (or (14), what is the same thing) is true. □

As a consequence, we obtain the third criterion of convergence.

Corollary 18. *Let $||\cdot||$ be a given submultiplicative norm on $\mathscr{L}(E,E)$, if there exists an n such that $J_n(A_0^T|E,A_1^T|E,||\cdot||)<1$, then the corresponding subdivision scheme is C^0.*

Proof. Let a matrix norm $||\cdot||$ on $\mathscr{L}(E,E)$ such that $J_n(A_0^T|E,A_1^T|E,||\cdot||)<1$ for a given $n>0$. There exists M such that $||A||_1 \leq M||A||$ for all $||A|| \in \mathscr{L}(E,E)$. It follows that

$$J_{nk}(A_0^T|E,A_1^T|E,||\cdot||_1) \leq MJ_n^k(A_0^T|E,A_1^T|E,||\cdot||) \quad \text{for all } k.$$

If k is large enough, $J_{nk}(A_0^T|E,A_1^T|E,||\cdot||_1)<1$. From (14), we obtain $\omega_h(S^{nk})<1$. From Theorem 3, the corresponding subdivision scheme is C^0. □

For the case of a period equal to 1, Corollary 18 has been proved by Daubechies and Lagarias [4, Theorem 2.2 and Lemma 2.3]. Our result is new for a period >1.

6 Numerical Results for a Specific Scheme

In this section, we numerically compare the three criteria of convergence for the subdivision scheme of quadratic interpolation around even integers.

Theorem 19. *Let $S:\ell_0(\mathbb{Z})\to\ell_0(\mathbb{Z})$ be a subdivision operator which is bounded with respect to the difference norm δ. We assume that the subdivision matrix is periodic of period p and has $[\sigma,\sigma']$ as support. Let $N=p-1+\sigma'-\sigma$, for $r=0,1$, we set B_r the restriction to E, the hyperplane $x_1+x_2+\ldots+x_N=0$, of the linear map whose matrix representation is $(s(j+r,i):i,j\in[-\sigma'+1,-\sigma+p-1])$. Then the three sequences $\sqrt[n]{\delta(S^n)}$, $\sqrt[n]{\omega_N(S^n)}$ and $\sqrt[n]{J_n(\{B_0,B_1\},||\cdot||)}$ (where $||\cdot||$ is any matrix norm on $\mathscr{L}(E,E)$) converge to the same value.* □

Proof. We recall a theorem which is proved by Akilov and Kantorovich [8, Theorem (2.V), p.165]. *If on a normed space with norm $||\cdot||$, U is a bounded linear operator, then the sequence $||U^n||^{1/n}$ converge to the number* $\inf_n\{||U^n||^{1/n}\}$. From that theorem, the sequence $\sqrt[n]{\delta(S^n)}$ converges. For the same reason, for any integer N, the limit of $\sqrt[n]{\omega_N(S^n)}$ as $n\to\infty$ exists. The norms δ and ω_N are equivalent: for all $f\in\ell_0(\mathbb{Z})$, $\delta(f)\leq\omega_N(f)\leq N\delta(f)$. It follows that for all n, $\delta(S^n)/N\leq\omega_N(S^n)\leq N\delta(S^n)$. From that, both sequences $\sqrt[n]{\delta(S^n)}$ and $\sqrt[n]{\omega_N(S^n)}$ has the same limit c.

We set $\mathscr{B}=\{B_0,B_1\}$. By Theorem 14, $J_n(\mathscr{B},||\cdot||_1)=\omega_N(S^n)$. Since the norms $||\cdot||$ and $||\cdot||_1$ of $\mathscr{L}(E,E)$ are equivalent, then both sequences $\sqrt[n]{J_n(\mathscr{B},||\cdot||)}$ and $\sqrt[n]{J_n(\mathscr{B},||\cdot||_1)}=\sqrt[n]{\omega_N(S^n)}$ has the same limit c. The three sequences $\sqrt[n]{\delta(S^n)}$, $\sqrt[n]{\omega_N(S^n)}$ and $\sqrt[n]{J_n(\mathscr{B},||\cdot||)}$ converge to c. □

Let us apply the previous theorem to the subdivision scheme $\mathscr{S}$ of quadratic interpolation around even integers. In this case, the period is 2, the support is $[-3,3]$ and $N=7$. In Table 1, we provide the ten first elements of the three sequences $\sqrt[n]{\delta(S^n)}$, $\sqrt[n]{\omega_7(S^n)}$ and $\sqrt[n]{J_n(\mathscr{B},||\cdot||_2)}$ where $||\cdot||_2$ is the euclidean norm. From Theorem 19, the sequences converge to the same value c. For each sequence, we put in bold the first element less than 1. These elements are $\delta(S)$, $\sqrt[3]{\omega_7(S^3)}$ and $\sqrt[5]{J_5(\mathscr{B},||\cdot||_2)}$. Each criterion shows that $\mathscr{S}$ is C^0.

We get a better approximation of the value c by computing $\delta(S^{20})^{1/20}=0.5358$. Since any quadratic function is a limit function of $\mathscr{S}$, from Theorem 3, we already have that $\delta(S^n)\geq 1/2^n$ for all $n>0$, this means that $c\geq 1/2$. By other means, it could be shown that for the subdivision scheme of quadratic interpolation around even integers, $c=1/2$ and that any limit function of the scheme is C^1.

Table 1. Norms of powers of a subdivision operator and joint norms

n	$\sqrt[n]{\delta(S^n)}$	$\sqrt[n]{\omega_7(S^n)}$	$\sqrt[n]{J_n(\mathscr{B}, \|\|\cdot\|\|_2)}$
1	**0.7500**	1.2500	2.7709
2	0.7500	1.1040	1.7068
3	0.6897	**0.9888**	1.2889
4	0.6580	0.9269	1.1087
5	0.6323	0.8550	**0.9759**
6	0.6141	0.8022	0.8918
7	0.5997	0.7598	0.8303
8	0.5884	0.7274	0.7850
9	0.5793	0.7014	0.7501
10	0.5717	0.6804	0.7225

7 Conclusion

We obtained three criteria of C^0-convergence of subdivision schemes for curves. Each one requires computation of norms, the respective norms being the difference norm, the modulus of continuity and the nth-joint norm. Criteria 1 and 2 can be applied to any affine nonuniform subdivision scheme and they are equivalent in the sense that if a subdivision scheme is C^0 convergent according to one criterion, it is also C^0 convergent according to the other. The third criterion can be used only for periodic affine schemes and in that case, Criteria 2 and 3 are equivalent. In spite of these equivalencies, the first criterion with the difference norm seems to be the most powerful from the computational point of view.

It would be interesting to find similar criteria of C^m-convergence, $m > 0$ of subdivision schemes for curves or even surfaces. For the study of the differentiability of order m for curves, the norm $||\Delta^{m+1} f||_\infty$ appears as a natural candidate. For surfaces, the choice of a suitable norm is less obvious and a good computation of norms of operators is a more difficult task. A few years ago, Charina, Conti and Sauer [2] investigated the L_p-convergence of multivariate scalar and vector subdivision schemes. The operator norm they used was very similar to our $\delta(S)$. In some sense, they anticipated our own work.

Acknowledgments

The author was supported by the Natural Sciences and Engineering Research Council of Canada. Moreover, we thank Professor Jean-Louis Merrien and Researcher Lucia Romani for their comments about two previous talks on the topic of this paper. We also thank both anonymous referees for making helpful comments and suggesting relevant references.

References

1. Buhmann, M.D., Micchelli, C.: Using two-slanted matrices for subdivision. Proc. London Math. Soc. 69(3), 428–448 (1994)
2. Charina, M., Conzi, C., Sauer, T.: L_p-convergence of subdivision schemes: joint spectral radius versus restricted spectral radius. In: Chui, C.K., Neamtu, M., Schumaker, L.L. (eds.) Approximation Theory, Gatlinburg, TN, 2004, pp. 129–150. Nashboro Press, Brentwood (2005)
3. Charina, M., Conti, C., Sauer, T.: Regularity of multivariate vector subdivision schemes. Numer. Algorithms 39, 97–113 (2005)
4. Daubechies, I., Lagarias, J.C.: Two-scale difference equations II. Local regularity, infinite products of matrices and fractals. SIAM J. Math. Anal. 22, 1031–1079 (1992)
5. Dubuc, S.: Scalar and Hermite subdivision schemes. Appl. Comp. Harm. Anal. 21, 376–394 (2006)
6. Dyn, N., Gregory, J.A., Levin, D.: Analysis of linear binary subdivision schemes for curve design. Constr. Approx. 7, 127–147 (1991)
7. Jia, R.-Q., Jiang, Q.: Spectral analysis of the transition operator and its application to smoothness anlysis of wavelets. SIAM J. Matrix Anal. Appl. 24, 1071–1109 (2003)
8. Kantorovic, L.V., Akilov, G.P.: Functional Analysis in Normed Spaces. Pergamon, Oxford (1964)
9. Rota, G.-C., Strang, W.G.: A note on the joint spectral radius. Indag. Math. 22, 379–381 (1960)

Geometric Design Using Space Curves with Rational Rotation-Minimizing Frames

Rida T. Farouki[1], Carlotta Giannelli[2], and Alessandra Sestini[3]

[1] Department of Mechanical and Aeronautical Engineering,
University of California, Davis, CA 95616, USA
farouki@ucdavis.edu
[2] Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze,
Viale Morgagni 65, 50134 Firenze, Italy
giannelli@dsi.unifi.it
[3] Dipartimento di Matematica "Ulisse Dini," Università degli Studi di Firenze,
Viale Morgagni 67a, 50134 Firenze, Italy
alessandra.sestini@unifi.it

Abstract. A rotation–minimizing adapted frame $(\mathbf{f}_1(t), \mathbf{f}_2(t), \mathbf{f}_3(t))$ on a given space curve $\mathbf{r}(t)$ is characterized by the fact that the frame vector $\mathbf{f}_1$ coincides with the tangent $\mathbf{t} = \mathbf{r}'/|\mathbf{r}'|$, while the frame angular velocity $\boldsymbol{\omega}$ maintains a zero component along it, i.e., $\boldsymbol{\omega} \cdot \mathbf{t} \equiv 0$. Such frames are useful in constructing swept surfaces and specifying the orientation of a rigid body moving along a given spatial path. Recently, the existence of quintic polynomial curves that have *rational* rotation–minimizing frames (quintic RRMF curves) has been demonstrated. These RRMF curves are necessarily Pythagorean–hodograph (PH) space curves, satisfying certain non–linear constraints among the complex coefficients of the Hopf map representation for spatial PH curves. Preliminary results on the design of quintic RRMF curves by the interpolation of G^1 spatial Hermite data are presented in this paper. This problem involves solving a non–linear system of equations in six complex unknowns. The solution is obtained by a semi–numerical scheme, in which the problem is reduced to computing positive real roots of a certain univariate polynomial. The quintic RRMF G^1 Hermite interpolants possess one residual angular degree of freedom, which can strongly influence the curve shape. Computed examples are included to illustrate the method and the resulting quintic RRMF curves.

Keywords: Pythagorean–hodograph space curves, rational rotation–minimizing frames, Hermite interpolation, Hopf map representation.

1 Introduction

In applications such as computer animation, motion control, and swept surface constructions, it is often necessary to specify the variation of an orthonormal frame along a curved path, that describes the orientation of a rigid body as it traverses this path. In typical cases, one frame vector is prescribed *a priori* (e.g., the unit tangent vector to the path, or the unit polar vector from a fixed origin to each point of the path) and only one degree of freedom remains, associated with the orientation of the two frame vectors in

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 194–208, 2010.

the plane orthogonal to this prescribed vector at each curve point. Frames that incorporate the unit tangent and unit polar vector as one component are known as *adapted* and *directed* curve frames, respectively [7]. An infinitude of such frames exists on any given space curve, but among them the *rotation–minimizing frames* (or RMFs) are of special interest. The distinctive property of RMFs is that their angular velocity always maintains a zero component in the direction of the prescribed frame vector.

The focus of this paper is on rotation–minimizing adapted frames — and, in particular, the construction of curves with *rational* rotation–minimizing frames (or RRMF curves). Specifically, the problem of constructing such curves through interpolation of G^1 Hermite data (end points and tangents) is addressed. The rational dependence of RMFs on the curve parameter is desirable on account of its exactitude, its compatibility with native CAD system representations, and the efficient subsequent computations it facilitates. The existence of non–degenerate quintic space curves with rational RMFs (or quintic RRMF curves) has recently been demonstrated in [9]. Such curves are necessarily Pythagorean–hodograph (PH) space curves, and the basis for the procedure described herein is the system of constraints [9] on the complex coefficients in the Hopf map representation of spatial PH curves, that characterizes RRMF curves.

The problem of Hermite interpolation by spatial PH curves typically admits a multiplicity of formal solutions. For example, the construction of C^1 spatial PH quintic Hermite interpolants incurs selection of two residual angular parameters [6, 8, 16]. Since spatial PH quintics must satisfy three scalar constraints in order to be RRMF curves, relaxing from C^1 to G^1 Hermite data results in a net loss of one residual degree of freedom — i.e., the quintic RRMF interpolants to spatial G^1 Hermite data comprise a *one–parameter family* of solutions.

The plan for this paper is as follows. After briefly reviewing the conditions for a PH space curve to have an RRMF in Section 2, the problem of G^1 Hermite interpolation using RRMF quintics is formulated in Section 3. A procedure for solving this problem is then described in Section 4, and a selection of computed examples is presented in Section 5. Finally, Section 6 summarizes the key results of this paper and makes some concluding remarks.

2 Rational RMFs on Polynomial Space Curves

An *adapted* frame on a regular parametric space curve $\mathbf{r}(t)$ is an orthonormal basis $(\mathbf{f}_1(t), \mathbf{f}_2(t), \mathbf{f}_3(t))$ for $\mathbb{R}^3$ such that $\mathbf{f}_1$ coincides with the unit tangent $\mathbf{t} = \mathbf{r}'/|\mathbf{r}'|$. The variation of such a frame may be characterized by its angular velocity $\boldsymbol{\omega}(t)$, such that

$$\frac{\mathrm{d}\mathbf{f}_1}{\mathrm{d}t} = \sigma\,\boldsymbol{\omega}\times\mathbf{f}_1\,, \qquad \frac{\mathrm{d}\mathbf{f}_2}{\mathrm{d}t} = \sigma\,\boldsymbol{\omega}\times\mathbf{f}_2\,, \qquad \frac{\mathrm{d}\mathbf{f}_3}{\mathrm{d}t} = \sigma\,\boldsymbol{\omega}\times\mathbf{f}_3\,,$$

$\sigma(t) = |\mathbf{r}'(t)|$ being the parametric speed of the curve. Among all adapted frames on a given space curve, the *rotation–minimizing frames* (RMFs) exhibit the least possible magnitude $|\boldsymbol{\omega}(t)|$ of the frame angular velocity [1, 12, 15]. An RMF may be characterized by the property that its angular velocity maintains a vanishing component in the direction of $\mathbf{f}_1 = \mathbf{t}$, i.e., $\boldsymbol{\omega}\cdot\mathbf{t} \equiv 0$, and hence the instantaneous rate of rotation of $\mathbf{f}_2$ and $\mathbf{f}_3$ about $\mathbf{f}_1$ is always zero.

In general, the RMFs of polynomial or rational curves do not admit rational dependence on the curve parameter, and this fact has prompted several authors to propose schemes for piecewise–rational RMF approximations [11,14,17,18]. Clearly, a polynomial space curve $\mathbf{r}(t) = (x(t), y(t), z(t))$ that admits a rational RMF — i.e., an *RRMF curve* — must have a *rational* unit tangent vector, and it must therefore be a Pythagorean–hodograph (PH) curve [5], whose derivative or hodograph $\mathbf{r}'(t) = (x'(t), y'(t), z'(t))$ satisfies

$$|\mathbf{r}'(t)|^2 = x'^2(t) + y'^2(t) + z'^2(t) = \sigma^2(t) \tag{1}$$

for a suitable polynomial $\sigma(t)$. Two useful (equivalent) representations for spatial PH curves, introduced in [3], are based on the algebra of quaternions $\mathbb{H}$ and the Hopf map, expressed in terms of complex number pairs $\mathbb{C} \times \mathbb{C}$.

Exact RMFs can be computed for any spatial PH curve [4], but in general they incur transcendental functions. The RRMF curves form a proper subset of all spatial PH curves. Using the quaternion model, it was shown in [13] that no RRMF cubics exist, except the degenerate cases of linear or planar curves. Subsequently, the existence of non–degenerate RRMF quintics was demonstrated in [9], using the Hopf map model, and conditions on the complex coefficients in this model were introduced, that identify a spatial PH quintic as an RRMF curve. The Hopf map $H : \mathbb{C}\times\mathbb{C} \to \mathbb{R}^3$ specifies a spatial Pythagorean hodograph in terms of two complex polynomials $\boldsymbol{\alpha}(t) = u(t) + \mathrm{i}\, v(t)$, $\boldsymbol{\beta}(t) = q(t) + \mathrm{i}\, p(t)$ as

$$\mathbf{r}'(t) = H(\boldsymbol{\alpha}(t), \boldsymbol{\beta}(t))\,, \tag{2}$$

where

$$H(\boldsymbol{\alpha}(t), \boldsymbol{\beta}(t)) = (\,|\boldsymbol{\alpha}(t)|^2 - |\boldsymbol{\beta}(t)|^2\,,\, 2\,\mathrm{Re}(\boldsymbol{\alpha}(t)\overline{\boldsymbol{\beta}}(t))\,,\, 2\,\mathrm{Im}(\boldsymbol{\alpha}(t)\overline{\boldsymbol{\beta}}(t))\,)\,. \tag{3}$$

In order to define an RRMF curve, the complex polynomials $\boldsymbol{\alpha}(t)$, $\boldsymbol{\beta}(t)$ in (3) must satisfy [9,13] the condition

$$\frac{\overline{\boldsymbol{\alpha}}\boldsymbol{\alpha}' - \overline{\boldsymbol{\alpha}}'\boldsymbol{\alpha} + \overline{\boldsymbol{\beta}}\boldsymbol{\beta}' - \overline{\boldsymbol{\beta}}'\boldsymbol{\beta}}{\overline{\boldsymbol{\alpha}}\boldsymbol{\alpha} + \overline{\boldsymbol{\beta}}\boldsymbol{\beta}} = \frac{\overline{\mathbf{w}}\mathbf{w}' - \overline{\mathbf{w}}'\mathbf{w}}{\overline{\mathbf{w}}\mathbf{w}}\,, \tag{4}$$

where $\mathbf{w}(t) = a(t) + \mathrm{i}\, b(t)$ is a complex polynomial with $\gcd(a(t), b(t)) = \text{constant}$. As noted in [9], when $\mathbf{w}(t)$ is either a real polynomial or a constant, the condition (4) becomes

$$\overline{\boldsymbol{\alpha}}\boldsymbol{\alpha}' - \overline{\boldsymbol{\alpha}}'\boldsymbol{\alpha} + \overline{\boldsymbol{\beta}}\boldsymbol{\beta}' - \overline{\boldsymbol{\beta}}'\boldsymbol{\beta} = 0\,.$$

This circumstance identifies coincidence of the RMF with the *Euler–Rodrigues frame* (ERF) — a rational adapted frame defined [2] for any spatial PH curve, that may be expressed in terms of the Hopf map model as

$$\begin{aligned}
\mathbf{e}_1(t) &= \frac{(\,|\boldsymbol{\alpha}|^2 - |\boldsymbol{\beta}|^2\,,\, 2\,\mathrm{Re}(\boldsymbol{\alpha}\overline{\boldsymbol{\beta}})\,,\, 2\,\mathrm{Im}(\boldsymbol{\alpha}\,\overline{\boldsymbol{\beta}})\,)}{|\boldsymbol{\alpha}|^2 + |\boldsymbol{\beta}|^2}\,,\\
\mathbf{e}_2(t) &= \frac{(-\,2\,\mathrm{Re}(\boldsymbol{\alpha}\,\boldsymbol{\beta})\,,\, \mathrm{Re}(\boldsymbol{\alpha}^2 - \boldsymbol{\beta}^2)\,,\, \mathrm{Im}(\boldsymbol{\alpha}^2 + \boldsymbol{\beta}^2)\,)}{|\boldsymbol{\alpha}|^2 + |\boldsymbol{\beta}|^2}\,,\\
\mathbf{e}_3(t) &= \frac{(\,2\,\mathrm{Im}(\boldsymbol{\alpha}\,\boldsymbol{\beta})\,,\, -\,\mathrm{Im}(\boldsymbol{\alpha}^2 - \boldsymbol{\beta}^2)\,,\, \mathrm{Re}(\boldsymbol{\alpha}^2 + \boldsymbol{\beta}^2)\,)}{|\boldsymbol{\alpha}|^2 + |\boldsymbol{\beta}|^2}\,.
\end{aligned} \tag{5}$$

It was shown in [2] that the ERF is not rotation–minimizing for any non–planar RRMF cubic or quintic. Nevertheless, the ERF serves as a useful rational adapted "reference" frame in the problem of identifying RRMF curves.

To define an RRMF quintic, we employ complex quadratic polynomials

$$\begin{aligned} \boldsymbol{\alpha}(t) &= \boldsymbol{\alpha}_0\,(1-t)^2 \,+\, \boldsymbol{\alpha}_1\,2(1-t)t \,+\, \boldsymbol{\alpha}_2\,t^2\,, \\ \boldsymbol{\beta}(t) &= \boldsymbol{\beta}_0\,(1-t)^2 \,+\, \boldsymbol{\beta}_1\,2(1-t)t \,+\, \boldsymbol{\beta}_2\,t^2\,, \end{aligned} \tag{6}$$

in (3), where $\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k \in \mathbb{C}$ for $k = 0, 1, 2$. The constraints on these six coefficients that identify non–degenerate RRMF quintics are

$$(|\boldsymbol{\alpha}_0|^2 + |\boldsymbol{\beta}_0|^2)\,|\overline{\boldsymbol{\alpha}}_1\boldsymbol{\alpha}_2 + \overline{\boldsymbol{\beta}}_1\boldsymbol{\beta}_2|^2 \;=\; (|\boldsymbol{\alpha}_2|^2 + |\boldsymbol{\beta}_2|^2)\,|\boldsymbol{\alpha}_0\overline{\boldsymbol{\alpha}}_1 + \boldsymbol{\beta}_0\overline{\boldsymbol{\beta}}_1|^2\,, \tag{7}$$

$$(|\boldsymbol{\alpha}_0|^2 + |\boldsymbol{\beta}_0|^2)\,(\boldsymbol{\alpha}_0\boldsymbol{\beta}_2 - \boldsymbol{\alpha}_2\boldsymbol{\beta}_0) \;=\; 2\,(\boldsymbol{\alpha}_0\overline{\boldsymbol{\alpha}}_1 + \boldsymbol{\beta}_0\overline{\boldsymbol{\beta}}_1)(\boldsymbol{\alpha}_0\boldsymbol{\beta}_1 - \boldsymbol{\alpha}_1\boldsymbol{\beta}_0)\,. \tag{8}$$

It has been shown in [9] that these are necessary and sufficient for the existence of a quadratic complex polynomial $\mathbf{w}(t)$ such that (4) is satisfied. Since (7) and (8) are constraints on *real* and *complex* values, respectively, they amount to *three* scalar constraints on the coefficients of the polynomials $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\beta}(t)$.

Once the coefficients of the two quadratic polynomials $\boldsymbol{\alpha}(t)$, $\boldsymbol{\beta}(t)$ satisfying (7) and (8) are known, the coefficients of the polynomial

$$\mathbf{w}(t) \;=\; \mathbf{w}_0\,(1-t)^2 \,+\, \mathbf{w}_1\,2(1-t)t \,+\, \mathbf{w}_2\,t^2$$

in (4) are given [9] in terms of them by

$$\mathbf{w}_0 \;=\; 1\,, \qquad \mathbf{w}_1 \;=\; \frac{\overline{\boldsymbol{\alpha}}_0\boldsymbol{\alpha}_1 + \overline{\boldsymbol{\beta}}_0\boldsymbol{\beta}_1}{|\boldsymbol{\alpha}_0|^2 + |\boldsymbol{\beta}_0|^2}\,, \qquad \mathbf{w}_2 \;=\; \frac{\overline{\boldsymbol{\alpha}}_1\boldsymbol{\alpha}_2 + \overline{\boldsymbol{\beta}}_1\boldsymbol{\beta}_2}{\boldsymbol{\alpha}_0\overline{\boldsymbol{\alpha}}_1 + \boldsymbol{\beta}_0\overline{\boldsymbol{\beta}}_1}\,.$$

The rational RMF $(\mathbf{f}_1(t), \mathbf{f}_2(t), \mathbf{f}_3(t))$ can then be expressed [9] in terms of the ERF (5) and the polynomials $a(t) = \mathrm{Re}(\mathbf{w}(t))$, $b(t) = \mathrm{Im}(\mathbf{w}(t))$ as

$$\begin{aligned} \mathbf{f}_1(t) &= \mathbf{e}_1(t)\,, \\ \mathbf{f}_2(t) &= \frac{a^2(t) - b^2(t)}{a^2(t) + b^2(t)}\,\mathbf{e}_2(t) \;-\; \frac{2\,a(t)b(t)}{a^2(t) + b^2(t)}\,\mathbf{e}_3(t)\,, \\ \mathbf{f}_3(t) &= \frac{2\,a(t)b(t)}{a^2(t) + b^2(t)}\,\mathbf{e}_2(t) \;+\; \frac{a^2(t) - b^2(t)}{a^2(t) + b^2(t)}\,\mathbf{e}_3(t)\,. \end{aligned} \tag{9}$$

This rational rotation–minimizing frame is of degree 8 in the curve parameter t.

3 Interpolation of G^1 Spatial Hermite Data

We are concerned with constructing quintic RRMF curves $\mathbf{r}(t)$ for $t \in [\,0, 1\,]$ that interpolate given G^1 spatial Hermite data — i.e., initial/final points $\mathbf{p}_i$, $\mathbf{p}_f$ and unit tangents $\mathbf{t}_i$, $\mathbf{t}_f$. Thus, the curve defined by (2)–(3) must satisfy (7)–(8) and

$$\mathbf{r}(0) = \mathbf{p}_i\,, \qquad \frac{\mathbf{r}'(0)}{|\mathbf{r}'(0)|} = \mathbf{t}_i\,, \qquad \mathbf{r}(1) = \mathbf{p}_f\,, \qquad \frac{\mathbf{r}'(1)}{|\mathbf{r}'(1)|} = \mathbf{t}_f\,. \tag{10}$$

The unit tangents $\mathbf{t}_i$, $\mathbf{t}_f$ can be specified in terms of polar angles θ_i, θ_f measured from the x–axis, and azimuthal angles ϕ_i, ϕ_f measured about the x–axis, as

$$\mathbf{t}_i \;=\; (\cos\theta_i, \sin\theta_i\cos\phi_i, \sin\theta_i\sin\phi_i)\,, \tag{11}$$

$$\mathbf{t}_f \;=\; (\cos\theta_f, \sin\theta_f\cos\phi_f, \sin\theta_f\sin\phi_f)\,. \tag{12}$$

Since the first condition in (10) is trivially satisfied by taking $\mathbf{p}_i$ as integration constant on integrating (3), we need only consider the displacement

$$\Delta\mathbf{p} \;=\; \int_0^1 \mathbf{r}'(t)\,\mathrm{d}t \;=\; \mathbf{p}_f - \mathbf{p}_i$$

rather than $\mathbf{p}_i$, $\mathbf{p}_f$ individually. Now it is always possible to choose a coordinate system in which the initial tangent $\mathbf{t}_i$ is in the (x,y)–plane, the initial point $\mathbf{p}_i$ is at the origin, and the displacement $\Delta\mathbf{p} = \mathbf{p}_f - \mathbf{p}_i$ lies on the x–axis. We say that such coordinates define *canonical Hermite data*, with $\phi_i = 0$ and $\Delta\mathbf{p} = (X,0,0)$. We henceforth assume data of this form, and for brevity we write $\phi_f = \phi$ and

$$(c_i, s_i) \;=\; (\cos\tfrac{1}{2}\theta_i, \sin\tfrac{1}{2}\theta_i)\,, \qquad (c_f, s_f) \;=\; (\cos\tfrac{1}{2}\theta_f, \sin\tfrac{1}{2}\theta_f)\,. \tag{13}$$

Note that, for non–planar data, θ_i, θ_f and ϕ must not be integer multiples of π.

Interpolation of the displacement $\Delta\mathbf{p} = (X,0,0)$ by a PH quintic yields the two conditions

$$\begin{aligned} 5\,X \;=\;& |\boldsymbol{\alpha}_0|^2 - |\boldsymbol{\beta}_0|^2 \\ &+\; \mathrm{Re}(\boldsymbol{\alpha}_0\overline{\boldsymbol{\alpha}}_1 - \boldsymbol{\beta}_0\overline{\boldsymbol{\beta}}_1) \\ &+\; \tfrac{1}{3}\,\mathrm{Re}(\boldsymbol{\alpha}_0\overline{\boldsymbol{\alpha}}_2 - \boldsymbol{\beta}_0\overline{\boldsymbol{\beta}}_2) \;+\; \tfrac{2}{3}\,(|\boldsymbol{\alpha}_1|^2 - |\boldsymbol{\beta}_1|^2) \\ &+\; \mathrm{Re}(\boldsymbol{\alpha}_1\overline{\boldsymbol{\alpha}}_2 - \boldsymbol{\beta}_1\overline{\boldsymbol{\beta}}_2) \\ &+\; |\boldsymbol{\alpha}_2|^2 - |\boldsymbol{\beta}_2|^2\,, \end{aligned} \tag{14}$$

$$\begin{aligned} 0 \;=\;& 2\,\boldsymbol{\alpha}_0\overline{\boldsymbol{\beta}}_0 \\ &+\; \boldsymbol{\alpha}_0\overline{\boldsymbol{\beta}}_1 + \boldsymbol{\alpha}_1\overline{\boldsymbol{\beta}}_0 \\ &+\; \tfrac{1}{3}\,(\boldsymbol{\alpha}_0\overline{\boldsymbol{\beta}}_2 + \boldsymbol{\alpha}_2\overline{\boldsymbol{\beta}}_0) \;+\; \tfrac{4}{3}\,\boldsymbol{\alpha}_1\overline{\boldsymbol{\beta}}_1 \\ &+\; \boldsymbol{\alpha}_1\overline{\boldsymbol{\beta}}_2 + \boldsymbol{\alpha}_2\overline{\boldsymbol{\beta}}_1 \\ &+\; 2\,\boldsymbol{\alpha}_2\overline{\boldsymbol{\beta}}_2\,. \end{aligned} \tag{15}$$

Since (14) is a scalar equation, while (15) is a relation among complex values, interpolating $\Delta\mathbf{p}$ incurs *three* scalar constraints on the coefficients of $\boldsymbol{\alpha}(t)$, $\boldsymbol{\beta}(t)$.

From (3) and (6), we can write the interpolation of end tangents as

$$\frac{\mathbf{r}'(0)}{|\mathbf{r}'(0)|} \;=\; \frac{(|\boldsymbol{\alpha}_0|^2 - |\boldsymbol{\beta}_0|^2,\, 2\,\mathrm{Re}(\boldsymbol{\alpha}_0\overline{\boldsymbol{\beta}}_0),\, 2\,\mathrm{Im}(\boldsymbol{\alpha}_0\overline{\boldsymbol{\beta}}_0))}{|\boldsymbol{\alpha}_0|^2 + |\boldsymbol{\beta}_0|^2} \;=\; \mathbf{t}_i\,, \tag{16}$$

$$\frac{\mathbf{r}'(1)}{|\mathbf{r}'(1)|} \;=\; \frac{(|\boldsymbol{\alpha}_2|^2 - |\boldsymbol{\beta}_2|^2,\, 2\,\mathrm{Re}(\boldsymbol{\alpha}_2\overline{\boldsymbol{\beta}}_2),\, 2\,\mathrm{Im}(\boldsymbol{\alpha}_2\overline{\boldsymbol{\beta}}_2))}{|\boldsymbol{\alpha}_2|^2 + |\boldsymbol{\beta}_2|^2} \;=\; \mathbf{t}_f\,. \tag{17}$$

Since $\mathbf{t}_i$ and $\mathbf{t}_f$ are *unit* vectors, (16) and (17) each yield two scalar constraints on the coefficients of $\boldsymbol{\alpha}(t)$, $\boldsymbol{\beta}(t)$. Hence, the G^1 Hermite interpolation conditions impose *seven* scalar constraints on the *six* complex coefficients of the polynomials $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\beta}(t)$. Thus, in conjunction with the RRMF constraints (7)–(8), we have altogether *ten* scalar constraints and *twelve* scalar unknowns.

Another scalar constraint may be imposed by noting [10] that the polynomials (6) embody one non–essential freedom. Namely, if the polynomials $\boldsymbol{\alpha}(t)$, $\boldsymbol{\beta}(t)$ generate a specific hodograph $\mathbf{r}'(t)$ through (3), the same hodograph is obtained on replacing them by $\exp(\mathrm{i}\xi)\,\boldsymbol{\alpha}(t)$, $\exp(\mathrm{i}\xi)\,\boldsymbol{\beta}(t)$ for any $\xi \in \mathbb{R}$. Hence we may assume, without loss of generality, that one of the coefficients in (6) is *real*. Thus, we expect the RRMF quintic interpolants to given spatial G^1 Hermite data to form a one–parameter family of space curves.

4 Solution of RRMF Hermite System

As noted in Sections 2 and 3, the interpolation of G^1 Hermite data by RRMF quintic curves involves a system of eleven scalar constraints on the six complex coefficients $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2$, leaving one scalar degree of freedom. Consider first interpolation of the end tangents.

Proposition 1. *For canonical–form Hermite data, interpolation of the two end tangents (11)–(12) may be achieved by expressing $\boldsymbol{\alpha}_0$, $\boldsymbol{\beta}_0$ and $\boldsymbol{\alpha}_2$, $\boldsymbol{\beta}_2$ in terms of complex values $\boldsymbol{\gamma}_0$ and $\boldsymbol{\gamma}_2$ as*

$$\boldsymbol{\alpha}_0 = \boldsymbol{\gamma}_0 c_i\,, \qquad \boldsymbol{\alpha}_2 = \boldsymbol{\gamma}_2 c_f \exp(\mathrm{i}\tfrac{1}{2}\phi)\,, \tag{18}$$

$$\boldsymbol{\beta}_0 = \boldsymbol{\gamma}_0 s_i\,, \qquad \boldsymbol{\beta}_2 = \boldsymbol{\gamma}_2 s_f \exp(-\mathrm{i}\tfrac{1}{2}\phi)\,. \tag{19}$$

Proof. Substituting (18)–(19) in (16)–(17) and simplifying yields (11)–(12). □

From (18) and (19) we see that

$$|\boldsymbol{\alpha}_0|^2 + |\boldsymbol{\beta}_0|^2 = |\boldsymbol{\gamma}_0|^2 \qquad \text{and} \qquad |\boldsymbol{\alpha}_2|^2 + |\boldsymbol{\beta}_2|^2 = |\boldsymbol{\gamma}_2|^2\,.$$

Thus, denoting by ρ^2 the ratio of the end–derivative magnitudes,

$$\rho^2 = \frac{|\mathbf{r}'(1)|}{|\mathbf{r}'(0)|} = \frac{|\boldsymbol{\alpha}_2|^2 + |\boldsymbol{\beta}_2|^2}{|\boldsymbol{\alpha}_0|^2 + |\boldsymbol{\beta}_0|^2}\,, \tag{20}$$

we have

$$|\boldsymbol{\gamma}_2| = \rho\,|\boldsymbol{\gamma}_0|\,.$$

Therefore, we can write $\boldsymbol{\gamma}_0 = \gamma \exp(\mathrm{i}\lambda_0)$ and $\boldsymbol{\gamma}_2 = \rho\,\gamma \exp(\mathrm{i}\lambda_2)$ where $\gamma, \rho \in \mathbb{R}^+$ and $\lambda_0, \lambda_2 \in [\,0, 2\pi\,]$. The redundancy of the representation (2) can be used to fix either λ_0 or λ_2. We choose $\lambda_0 = 0$ and, for simplicity, set $\lambda_2 = \lambda$. Invoking the notations (13), we obtain the expressions

$$\boldsymbol{\alpha}_0 = \gamma\, c_i\,, \qquad \boldsymbol{\alpha}_2 = \rho\,\gamma\, c_f \exp(\mathrm{i}(\lambda + \tfrac{1}{2}\phi))\,, \tag{21}$$

$$\boldsymbol{\beta}_0 = \gamma\, s_i\,, \qquad \boldsymbol{\beta}_2 = \rho\,\gamma\, s_f \exp(\mathrm{i}(\lambda - \tfrac{1}{2}\phi))\,, \tag{22}$$

where $c_i, s_i, c_f, s_f \neq 0$ and ϕ is not an integer multiple of π for non–planar data.

Consider now the RRMF conditions (7)–(8). If we substitute the expressions for $\boldsymbol{\alpha}_0, \boldsymbol{\beta}_0, \boldsymbol{\alpha}_2, \boldsymbol{\beta}_2$ and simplify, we obtain

$$| \, c_f \exp(\mathrm{i}\tfrac{1}{2}\phi) \, \overline{\boldsymbol{\alpha}}_1 + s_f \exp(-\mathrm{i}\tfrac{1}{2}\phi) \, \overline{\boldsymbol{\beta}}_1 \, | \;=\; | \, c_i \, \overline{\boldsymbol{\alpha}}_1 + s_i \, \overline{\boldsymbol{\beta}}_1 \, | \, , \tag{23}$$

$$\begin{aligned} \rho \, \gamma^2 \exp(\mathrm{i}\lambda) \, [\, c_i s_f \exp(-\mathrm{i}\tfrac{1}{2}\phi) - c_f s_i \exp(\mathrm{i}\tfrac{1}{2}\phi) \,] & \\ = \; 2(c_i \overline{\boldsymbol{\alpha}}_1 + s_i \overline{\boldsymbol{\beta}}_1)(c_i \boldsymbol{\beta}_1 - s_i \boldsymbol{\alpha}_1) \, . & \end{aligned} \tag{24}$$

Condition (23) implies that, for some angular parameter η, we must have

$$\exp(\mathrm{i}\eta) \, [\, c_f \exp(\mathrm{i}\tfrac{1}{2}\phi) \, \overline{\boldsymbol{\alpha}}_1 + s_f \exp(-\mathrm{i}\tfrac{1}{2}\phi) \, \overline{\boldsymbol{\beta}}_1 \,] \;=\; c_i \, \overline{\boldsymbol{\alpha}}_1 + s_i \, \overline{\boldsymbol{\beta}}_1 \, ,$$

and hence

$$[\, s_f \exp(\mathrm{i}\tfrac{1}{2}\phi) - s_i \exp(\mathrm{i}\eta) \,] \, \boldsymbol{\beta}_1 \;=\; [\, c_i \exp(\mathrm{i}\eta) - c_f \exp(-\mathrm{i}\tfrac{1}{2}\phi) \,] \, \boldsymbol{\alpha}_1 \, . \tag{25}$$

Now if η is such that $s_f \exp(\mathrm{i}\frac{1}{2}\phi) = s_i \exp(\mathrm{i}\eta)$, condition (25) implies that $\boldsymbol{\alpha}_1 = 0$. This is possible only if $s_f = s_i$ and $\eta = \frac{1}{2}\phi$, or $s_f = -s_i$ and $\eta = \frac{1}{2}\phi + \pi$. In the former case (24) is satisfied by taking $\lambda = \frac{1}{2}\pi$ or $\frac{3}{2}\pi$ and $|\boldsymbol{\beta}_1|^2 = \rho \, \gamma^2 |\sin \frac{1}{2}\phi|$. In the latter case (24) is satisfied with $\lambda = 0$ or π and $|\boldsymbol{\beta}_1|^2 = \rho \, \gamma^2 |\cos \frac{1}{2}\phi|$. In all other cases (25) implies that

$$\boldsymbol{\beta}_1 \;=\; \frac{c_i \exp(\mathrm{i}\eta) - c_f \exp(-\mathrm{i}\tfrac{1}{2}\phi)}{s_f \exp(\mathrm{i}\tfrac{1}{2}\phi) - s_i \exp(\mathrm{i}\eta)} \, \boldsymbol{\alpha}_1 \, . \tag{26}$$

Substituting (26) into (24), cancelling the factor[1] $c_i s_f \exp(-\mathrm{i}\frac{1}{2}\phi) - c_f s_i \exp(\mathrm{i}\frac{1}{2}\phi)$ from both sides, and setting $\boldsymbol{\epsilon} = c_i c_f \exp(-\mathrm{i}\frac{1}{2}\phi) + s_i s_f \exp(\mathrm{i}\frac{1}{2}\phi)$, we obtain

$$|\boldsymbol{\alpha}_1|^2 \;=\; \tfrac{1}{2} \, \rho \, \gamma^2 \, \frac{| \, s_f \exp(\mathrm{i}\frac{1}{2}\phi) - s_i \exp(\mathrm{i}\eta) \, |^2}{\exp(-\mathrm{i}\lambda) \, [\exp(\mathrm{i}\eta) - \boldsymbol{\epsilon} \,]} \, . \tag{27}$$

For a valid solution, the expression on the right must have a non–negative real value. This is equivalent to satisfaction of the relations

$$\mathrm{Im}(\exp(-\mathrm{i}\lambda) \, [\exp(\mathrm{i}\eta) - \boldsymbol{\epsilon} \,]) \;=\; 0 \, , \qquad \mathrm{Re}(\exp(-\mathrm{i}\lambda) \, [\exp(\mathrm{i}\eta) - \boldsymbol{\epsilon} \,]) \;>\; 0 \, ,$$

by the parameters η and λ, and consequently

$$\exp(-\mathrm{i}\lambda) \;=\; \boldsymbol{\mu}_0(\eta) \qquad \text{with} \qquad \boldsymbol{\mu}_0(\eta) \;=\; \frac{\exp(-\mathrm{i}\eta) - \overline{\boldsymbol{\epsilon}}}{| \exp(-\mathrm{i}\eta) - \overline{\boldsymbol{\epsilon}} \, |} \, . \tag{28}$$

Substituting from (28) into (27), we obtain

$$|\boldsymbol{\alpha}_1|^2 \;=\; \rho \, \gamma^2 f_1(\eta) \qquad \text{with} \qquad f_1(\eta) \;=\; \frac{\frac{1}{2} \, | \, s_f \exp(\mathrm{i}\frac{1}{2}\phi) - s_i \exp(\mathrm{i}\eta) \, |^2}{| \exp(\mathrm{i}\eta) - \boldsymbol{\epsilon} \, |} \, . \tag{29}$$

[1] Under the stated assumptions on c_i, s_i, c_f, s_f and ϕ, this factor is always non–zero. The same is true of the term $\exp(\mathrm{i}\eta) - \boldsymbol{\epsilon}$, for all values of η.

Assuming that[2] $3\overline{\boldsymbol{\alpha}}_0 + 4\overline{\boldsymbol{\alpha}}_1 + 3\overline{\boldsymbol{\alpha}}_2 \neq 0$, equation (15) allows $\boldsymbol{\beta}_1$ to be expressed in terms of $\boldsymbol{\alpha}_1$ as

$$\boldsymbol{\beta}_1 \;=\; -\,\frac{(6\boldsymbol{\beta}_0 + \boldsymbol{\beta}_2)\,\overline{\boldsymbol{\alpha}}_0 + 3(\boldsymbol{\beta}_0 + \boldsymbol{\beta}_2)\,\overline{\boldsymbol{\alpha}}_1 + (\boldsymbol{\beta}_0 + 6\boldsymbol{\beta}_2)\,\overline{\boldsymbol{\alpha}}_2}{3\overline{\boldsymbol{\alpha}}_0 + 4\overline{\boldsymbol{\alpha}}_1 + 3\overline{\boldsymbol{\alpha}}_2}\,.$$

Equating this expression for $\boldsymbol{\beta}_1$ with (26) and cancelling γ from both sides gives the equation

$$\boldsymbol{\delta}_0(\eta,\rho)\,\boldsymbol{\alpha}_1 \;+\; \boldsymbol{\delta}_1(\eta,\rho)\,\overline{\boldsymbol{\alpha}}_1 \;=\; \gamma\,\boldsymbol{\delta}_2(\eta,\rho)\,, \tag{30}$$

for $\boldsymbol{\alpha}_1$, with coefficients

$$\begin{aligned}
\boldsymbol{\delta}_0(\eta,\rho) \;&=\; 3\,[\,c_i + \rho\,c_f\boldsymbol{\mu}_0\exp(-\mathrm{i}\tfrac{1}{2}\phi)\,]\,\boldsymbol{\mu}_1\,,\\
\boldsymbol{\delta}_1(\eta,\rho) \;&=\; 3\,[\,s_i + \rho\,s_f\overline{\boldsymbol{\mu}}_0\exp(-\mathrm{i}\tfrac{1}{2}\phi)\,]\,,\\
\boldsymbol{\delta}_2(\eta,\rho) \;&=\; -\,6\,[\,c_is_i + \rho^2 c_f s_f\exp(-\mathrm{i}\phi)\,] - 4\,\rho\,f_1\,\boldsymbol{\mu}_1\\
&\qquad -\,\rho\,\exp(-\mathrm{i}\tfrac{1}{2}\phi)[\,c_is_f\,\overline{\boldsymbol{\mu}}_0 + c_f s_i\,\boldsymbol{\mu}_0\,]\,,
\end{aligned} \tag{31}$$

where we define

$$\boldsymbol{\mu}_1(\eta) \;=\; \frac{c_i\exp(\mathrm{i}\eta) - c_f\exp(-\mathrm{i}\frac{1}{2}\phi)}{s_f\exp(\mathrm{i}\frac{1}{2}\phi) - s_i\exp(\mathrm{i}\eta)}\,. \tag{32}$$

Now the pairs of (ρ,η) values such that $|\boldsymbol{\delta}_0|^2 = |\boldsymbol{\delta}_1|^2$ and $\mathrm{Re}((\boldsymbol{\delta}_0 - \boldsymbol{\delta}_1)\overline{\boldsymbol{\delta}}_2) \neq 0$ or $\mathrm{Im}((\boldsymbol{\delta}_0 + \boldsymbol{\delta}_1)\overline{\boldsymbol{\delta}}_2) \neq 0$ are unacceptable, because equation (30) has no solution. On the other hand, if there exist pairs such that $|\boldsymbol{\delta}_0|^2 = |\boldsymbol{\delta}_1|^2$ and $\mathrm{Re}((\boldsymbol{\delta}_0 - \boldsymbol{\delta}_1)\overline{\boldsymbol{\delta}}_2) = \mathrm{Im}((\boldsymbol{\delta}_0 + \boldsymbol{\delta}_1)\overline{\boldsymbol{\delta}}_2) = 0$, equation (30) corresponds to only one scalar condition. In this special case, we also need equation (29) to determine $\boldsymbol{\alpha}_1$.

In general, we can assume that $|\boldsymbol{\delta}_0|^2 \neq |\boldsymbol{\delta}_1|^2$ and derive $\boldsymbol{\alpha}_1$ from (30) as

$$\boldsymbol{\alpha}_1 \;=\; \gamma\,\frac{\overline{\boldsymbol{\delta}}_0\boldsymbol{\delta}_2 - \boldsymbol{\delta}_1\overline{\boldsymbol{\delta}}_2}{|\boldsymbol{\delta}_0|^2 - |\boldsymbol{\delta}_1|^2}\,.$$

Once η is chosen, the correct ρ value is identified via (29) by the positive real roots of the equation

$$|\,\overline{\boldsymbol{\delta}}_0\boldsymbol{\delta}_2 - \boldsymbol{\delta}_1\overline{\boldsymbol{\delta}}_2\,|^2 \;-\; \rho\,f_1(\eta)\,(|\boldsymbol{\delta}_0|^2 - |\boldsymbol{\delta}_1|^2)^2 \;=\; 0\,, \tag{33}$$

which is a polynomial equation of degree 6 with real coefficients. We may write

$$\overline{\boldsymbol{\delta}}_0\boldsymbol{\delta}_2 - \boldsymbol{\delta}_1\overline{\boldsymbol{\delta}}_2 \;=\; \sum_{k=0}^{3}\mathbf{z}_k\,\rho^k\,,$$

with

$$\begin{aligned}
\mathbf{z}_0 \;&=\; 18\,c_is_i(s_i - c_i\overline{\boldsymbol{\mu}}_1)\,,\\
\mathbf{z}_1 \;&=\; 18\,c_is_i\overline{\boldsymbol{\mu}}_0[s_f\exp(-\mathrm{i}\tfrac{1}{2}\phi) - c_f\overline{\boldsymbol{\mu}}_1\exp(\mathrm{i}\tfrac{1}{2}\phi)] + 3\,s_i\exp(\mathrm{i}\tfrac{1}{2}\phi)(c_is_f\boldsymbol{\mu}_0 + c_fs_i\overline{\boldsymbol{\mu}}_0)\\
&\quad + 12\,f_1\overline{\boldsymbol{\mu}}_1(s_i - c_i\boldsymbol{\mu}_1) - 3\,c_i\overline{\boldsymbol{\mu}}_1\exp(-\mathrm{i}\tfrac{1}{2}\phi)(c_is_f\overline{\boldsymbol{\mu}}_0 + c_fs_i\boldsymbol{\mu}_0)\,,\\
\mathbf{z}_2 \;&=\; 18\,c_fs_f[s_i\exp(\mathrm{i}\phi) - c_i\overline{\boldsymbol{\mu}}_1\exp(-\mathrm{i}\phi)] + 3\,s_f\overline{\boldsymbol{\mu}}_0(c_is_f\boldsymbol{\mu}_0 + c_fs_i\overline{\boldsymbol{\mu}}_0))\\
&\quad + 12\,f_1\overline{\boldsymbol{\mu}}_0\overline{\boldsymbol{\mu}}_1[s_f\exp(-\mathrm{i}\tfrac{1}{2}\phi) - c_f\boldsymbol{\mu}_1\exp(\mathrm{i}\tfrac{1}{2}\phi)] - 3\,c_f\overline{\boldsymbol{\mu}}_1\overline{\boldsymbol{\mu}}_0(c_is_f\overline{\boldsymbol{\mu}}_0 + c_fs_i\boldsymbol{\mu}_0)\,,\\
\mathbf{z}_3 \;&=\; 18\,c_fs_f\overline{\boldsymbol{\mu}}_0[s_f\exp(\mathrm{i}\tfrac{1}{2}\phi) - c_f\overline{\boldsymbol{\mu}}_1\exp(-\mathrm{i}\tfrac{1}{2}\phi)]\,,
\end{aligned}$$

[2] This incurs no loss of generality: the condition $\boldsymbol{\alpha}_1 = -\frac{3}{4}\,(\boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_2)$ imposes two more scalar constraints on η and ρ, so no degrees of freedom remain for satisfying (27).

while

$$|\boldsymbol{\delta}_0|^2 \;=\; \sum_{k=0}^{2} d_{0k}\rho^k \qquad \text{and} \qquad |\boldsymbol{\delta}_1|^2 \;=\; \sum_{k=0}^{2} d_{1k}\rho^k \,,$$

with

$$\begin{aligned}
d_{00} &= 9\,|\boldsymbol{\mu}_1|^2 c_i^2 \,, \\
d_{01} &= 9\,|\boldsymbol{\mu}_1|^2 c_i c_f \,[\,\boldsymbol{\mu}_0 \exp(-\mathrm{i}\tfrac{1}{2}\phi) + \overline{\boldsymbol{\mu}}_0 \exp(\mathrm{i}\tfrac{1}{2}\phi)\,] \,, \\
d_{02} &= 9\,|\boldsymbol{\mu}_1|^2 c_f^2 \,, \\
\\
d_{10} &= 9\,s_i^2 \,, \\
d_{11} &= 9\,s_i s_f \,[\,\overline{\boldsymbol{\mu}}_0 \exp(-\mathrm{i}\tfrac{1}{2}\phi) + \boldsymbol{\mu}_0 \exp(\mathrm{i}\tfrac{1}{2}\phi)\,] \,, \\
d_{12} &= 9\,s_f^2 \,.
\end{aligned}$$

Hence, setting

$$|\,\overline{\boldsymbol{\delta}}_0\boldsymbol{\delta}_2 - \boldsymbol{\delta}_1\overline{\boldsymbol{\delta}}_2\,|^2 \;=\; \sum_{k=0}^{6} c_k\rho^k \,,$$

where

$$\begin{aligned}
c_0 &= |\mathbf{z}_0|^2 \,, \\
c_1 &= 2\,\mathrm{Re}(\mathbf{z}_1\overline{\mathbf{z}}_0) \,, \\
c_2 &= 2\,\mathrm{Re}(\mathbf{z}_2\overline{\mathbf{z}}_0) + |\mathbf{z}_1|^2 \,, \\
c_3 &= 2\,\mathrm{Re}(\mathbf{z}_3\overline{\mathbf{z}}_0) + 2\,\mathrm{Re}(\mathbf{z}_2\overline{\mathbf{z}}_1) \,, \\
c_4 &= 2\,\mathrm{Re}(\mathbf{z}_3\overline{\mathbf{z}}_1) + |\mathbf{z}_2|^2 \,, \\
c_5 &= 2\,\mathrm{Re}(\mathbf{z}_3\overline{\mathbf{z}}_2) \,, \\
c_6 &= |\mathbf{z}_3|^2 \,,
\end{aligned} \tag{34}$$

equation (33) reduces to

$$c_6\,\rho^6 \;+\; \sum_{j=1}^{5}(c_j - e_{j-1}f_1)\rho^j \;+\; c_0 \;=\; 0 \,, \tag{35}$$

with

$$\begin{aligned}
e_0 &= (d_{00} - d_{10})^2 \,, \\
e_1 &= 2\,(d_{00} - d_{10})(d_{01} - d_{11}) \,, \\
e_2 &= (d_{01} - d_{11})^2 + 2\,(d_{00} - d_{10})(d_{02} - d_{12}) \,, \\
e_3 &= 2\,(d_{01} - d_{11})(d_{02} - d_{12}) \,, \\
e_4 &= (d_{02} - d_{12})^2 \,.
\end{aligned}$$

Equation (35) must possess at least one positive real root, for some value of the angular parameter η, if an RRMF quintic G^1 Hermite interpolant is to exist. Since the coefficients of (35) have a complicated, non–linear dependence on the Hermite data and on

the parameter η, a thorough investigation of the existence (and number) of interpolants is a challenging task, beyond our present scope. We hope to address it in a future study.

From (21)–(22) and (26)–(27) we observe that the coefficients of $\boldsymbol{\alpha}(t)$, $\boldsymbol{\beta}(t)$ are all proportional to γ. Hence, we may write

$$\boldsymbol{\alpha}_k = \gamma\, \mathbf{a}_k \qquad \text{and} \qquad \boldsymbol{\beta}_k = \gamma\, \mathbf{b}_k \tag{36}$$

for $k = 0, 1, 2$, where

$$\mathbf{a}_0 \;=\; c_i\,, \qquad \mathbf{a}_1 \;=\; \frac{\overline{\boldsymbol{\delta}}_0\boldsymbol{\delta}_2 - \boldsymbol{\delta}_1\overline{\boldsymbol{\delta}}_2}{|\boldsymbol{\delta}_0|^2 - |\boldsymbol{\delta}_1|^2}\,, \qquad \mathbf{a}_2 \;=\; \rho\, c_f\, \overline{\boldsymbol{\mu}}_0 \exp(\mathrm{i}\tfrac{1}{2}\phi)\,, \tag{37}$$

$$\mathbf{b}_0 \;=\; s_i\,, \qquad \mathbf{b}_1 \;=\; \boldsymbol{\mu}_1\mathbf{a}_1\,, \qquad \mathbf{b}_2 \;=\; \rho\, s_f\, \overline{\boldsymbol{\mu}}_0 \exp(-\mathrm{i}\tfrac{1}{2}\phi)\,. \tag{38}$$

Substituting (36) into (14), the γ value corresponding to a positive real root of (35) can be computed as

$$\gamma \;=\; \sqrt{\frac{5X}{f_2(\eta)}}\,, \tag{39}$$

where $f_2(\eta)$ is defined by

$$\begin{aligned} f_2(\eta) \;=\; & |\mathbf{a}_0|^2 - |\mathbf{b}_0|^2 \;+\; \mathrm{Re}(\mathbf{a}_0\overline{\mathbf{a}}_1 - \mathbf{b}_0\overline{\mathbf{b}}_1) \\ & +\tfrac{1}{3}\,\mathrm{Re}(\mathbf{a}_0\overline{\mathbf{a}}_2 - \mathbf{b}_0\overline{\mathbf{b}}_2) \;+\; \tfrac{2}{3}\,(|\mathbf{a}_1|^2 - |\mathbf{b}_1|^2) \\ & +\mathrm{Re}(\mathbf{a}_1\overline{\mathbf{a}}_2 - \mathbf{b}_1\overline{\mathbf{b}}_2) \;+\; |\mathbf{a}_2|^2 - |\mathbf{b}_2|^2\,. \end{aligned} \tag{40}$$

Of course, we must require $f_2(\eta) > 0$ for (39) to yield a real γ value. Again, due to the complicated dependence of (40) on η and the prescribed Hermite data, a detailed study of the conditions under which this holds is deferred to a future study. For the present, we only observe from experience with experimental tests that there are infinitely many admissible choices for η in the case of sufficiently dense data sampled from a smooth analytic curve, some of which produce very reasonable shapes. On the other hand, as shown in the examples of the following section, we have also been able to identify admissible fair–shaped RRMF quintic interpolants for many other data sets.

We conclude by summarizing the computation of RRMF quintic interpolants to spatial G^1 Hermite data as follows (for brevity, we exclude the cases $\eta = \frac{1}{2}\phi$ when $s_f = s_i$, and $\eta = \frac{1}{2}\phi + \pi$ when $s_f = -s_i$). The procedure employs N_η uniformly–sampled values of the η parameter.

Algorithm

Input: $\mathbf{p}_i, \mathbf{p}_f, \mathbf{t}_i, \mathbf{t}_f, N_\eta$

1. transform the Hermite data to canonical form;
2. determine $\theta_i, \theta_f, \phi = \phi_f$ from expressions (11)–(12);
3. compute c_i, s_i and c_f, s_f from (13);
4. compute $\boldsymbol{\epsilon} = c_i c_f \exp(-\mathrm{i}\frac{1}{2}\phi) + s_i s_f \exp(\mathrm{i}\frac{1}{2}\phi)$;
5. for $\eta = 2\pi k/N_\eta$ with $k = 0, \ldots, N_\eta - 1$:

(a) compute $\boldsymbol{\mu}_0$, $\boldsymbol{\mu}_1$, f_1 from (28), (32), (29);
(b) compute $c_0, \ldots, c_6$ from (34);
(c) find a positive real root ρ of equation (35);
if no positive real root exists, return to step 5;
(d) compute $\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \boldsymbol{\delta}_2$ from (31);
(e) compute $\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2$ and $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$ from (37) and (38);
(f) compute f_2 from (40) — if $f_2 \leq 0$ return to step 5;
(g) determine γ from expression (39);
(h) compute $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2$ and $\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2$ from (36)–(38);
(i) construct the hodograph (2) from $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\beta}(t)$;
(j) transform to original coordinates by inverting step 1.

Output: a set of RRMF quintics interpolating the Hermite data, corresponding to the chosen η values.

5 Numerical Results

The numerical results show that the η, ρ values can significantly influence the shape of the resulting RRMF interpolant. In computing the following examples, we used the MATLAB function `roots` to solve (35). The Bernstein coefficients of the polynomials $\boldsymbol{\alpha}(t)$, $\boldsymbol{\beta}(t)$, $\mathbf{w}(t)$ are quoted to five significant digits.

Example 1. Figure 1 shows two RRMF quintic interpolants to the data

$$\mathbf{p}_0 = (0,0,0)\,, \qquad \mathbf{p}_1 = (1,1,1)\,, \qquad \mathbf{t}_0 = \frac{(1,0,1)}{\sqrt{2}}\,, \qquad \mathbf{t}_1 = \frac{(0,1,1)}{\sqrt{2}}\,,$$

together with their Bézier control polygons. After transforming this data to canonical form, the coefficients of the $\boldsymbol{\alpha}(t)$, $\boldsymbol{\beta}(t)$, $\mathbf{w}(t)$ polynomials are

$$\begin{aligned}
&\boldsymbol{\alpha}_0 = 1.4194\,, && \boldsymbol{\alpha}_1 = -0.7920 + 0.4058\,\mathrm{i}\,, && \boldsymbol{\alpha}_2 = -0.9158 + 2.5605\,\mathrm{i}\,,\\
&\boldsymbol{\beta}_0 = 0.4512\,, && \boldsymbol{\beta}_1 = 1.1392 + 0.7361\,\mathrm{i}\,, && \boldsymbol{\beta}_2 = -0.5593 - 0.6590\,\mathrm{i}\,,\\
&\mathbf{w}_0 = 1.0\,, && \mathbf{w}_1 = -0.2751 + 0.4094\,\mathrm{i}\,, && \mathbf{w}_2 = 1.1863 + 1.5044\,\mathrm{i}\,,
\end{aligned}$$

for the curve on the left with $(\eta, \rho) = (5.2000, 1.9158)$, and

$$\begin{aligned}
&\boldsymbol{\alpha}_0 = 1.5363\,, && \boldsymbol{\alpha}_1 = 1.1372 + 0.4334\,\mathrm{i}\,, && \boldsymbol{\alpha}_2 = 0.7595 + 1.2735\,\mathrm{i}\,,\\
&\boldsymbol{\beta}_0 = 0.4883\,, && \boldsymbol{\beta}_1 = -0.0461 - 0.2865\,\mathrm{i}\,, && \boldsymbol{\beta}_2 = -0.4712 + 0.0067\,\mathrm{i}\,,\\
&\mathbf{w}_0 = 1.0\,, && \mathbf{w}_1 = 0.6637 + 0.2024\,\mathrm{i}\,, && \mathbf{w}_2 = 0.6024 + 0.7542\,\mathrm{i}\,,
\end{aligned}$$

for the curve on the right with $(\eta, \rho) = (4.3250, 0.9652)$. Both cases satisfy (4).

The corresponding values of the shape integrals[3] (89) in [8] are

$$L = 2.3259\,, \quad E = 44.509\,, \quad E_{\mathrm{rmf}} = 22.856\,,$$

[3] L is the total arc length; E is the elastic energy associated with the Frenet frame; and E_{rmf} is the energy associated with the RMF.

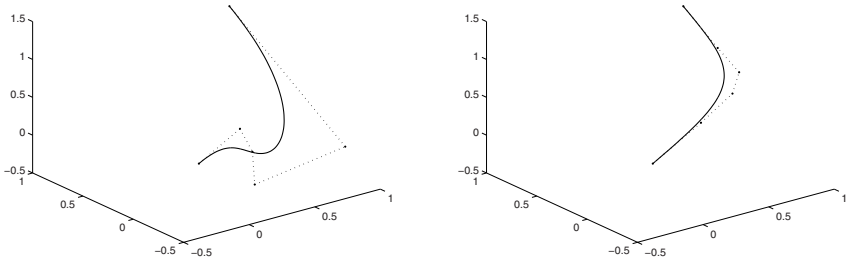


Fig. 1. The RRMF quintic interpolants of Example 1, with Bézier control polygons

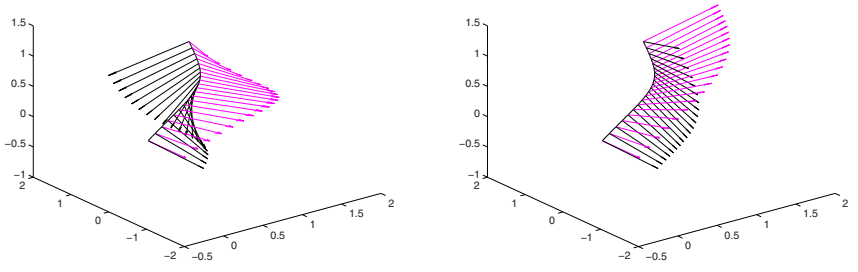


Fig. 2. Comparison of the ERF (left) and RMF (right) along the RRMF quintic shown on the right in Figure 1. For clarity, the unit tangent vector is omitted from the plots.

for the curve on the left, and

$$L = 1.9070\,, \quad E = 5.7495\,, \quad E_{\mathrm{rmf}} = 1.4641\,,$$

for the curve on the right. Figure 2 shows the variation of the ERF and RMF along the RRMF quintic on the right in Figure 1.

Example 2. Figure 3 shows two RRMF quintic interpolants to the data

$$\mathbf{p}_0 = (0,0,0)\,, \qquad \mathbf{p}_1 = (1,0,0)\,, \qquad \mathbf{t}_0 = \frac{(1,1,0)}{\sqrt{2}}\,, \qquad \mathbf{t}_1 = \frac{\mathbf{d}_1}{|\mathbf{d}_1|}\,,$$

where $\mathbf{d}_1 = (0.2, 0.2, 0.4057)$, together with their Bézier control polygons.

In this case, the coefficients of the $\boldsymbol{\alpha}(t)$, $\boldsymbol{\beta}(t)$, $\mathbf{w}(t)$ polynomials are

$$\begin{aligned}
&\boldsymbol{\alpha}_0 = 1.9240\,, && \boldsymbol{\alpha}_1 = 0.3403 - 0.9857\,\mathrm{i}\,, && \boldsymbol{\alpha}_2 = -0.8882 - 1.2811\,\mathrm{i}\,,\\
&\boldsymbol{\beta}_0 = 0.7970\,, && \boldsymbol{\beta}_1 = 0.1805 + 1.4013\,\mathrm{i}\,, && \boldsymbol{\beta}_2 = -1.0041 + 0.1499\,\mathrm{i}\,,\\
&\mathbf{w}_0 = 1.0\,, && \mathbf{w}_1 = 0.1841 - 0.1798\,\mathrm{i}\,, && \mathbf{w}_2 = 0.7110 - 0.5408\,\mathrm{i}\,,
\end{aligned}$$

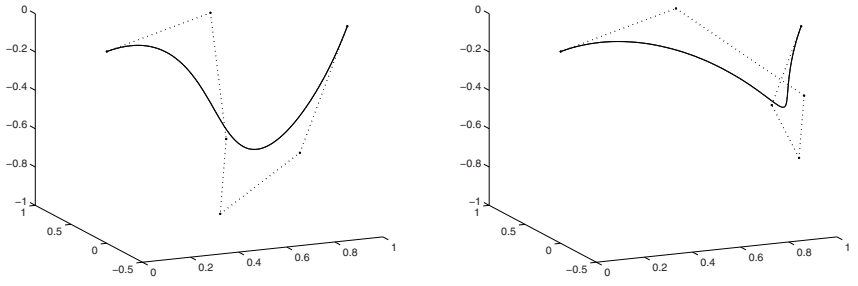


Fig. 3. The RRMF quintic interpolants of Example 2, with Bézier control polygons

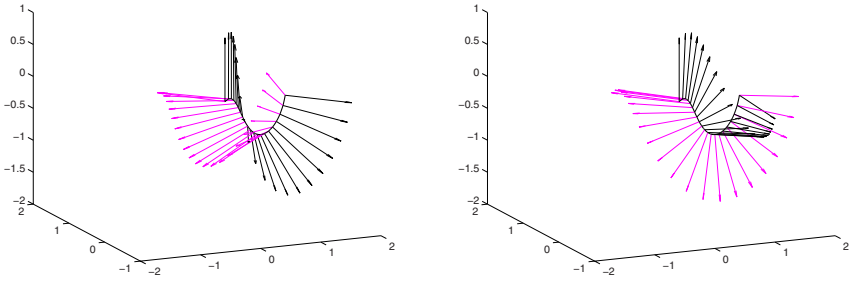


Fig. 4. Comparison of the ERF (left) and RMF (right) along the RRMF quintic shown on the left in Figure 3. For clarity, the unit tangent vector is omitted in these plots.

for the curve on the left with $(\eta, \rho) = (4.2000, 0.8933)$, and

$$\begin{aligned} \boldsymbol{\alpha}_0 &= 2.0292\,, & \boldsymbol{\alpha}_1 &= 0.8559 - 0.4150\,\mathrm{i}\,, & \boldsymbol{\alpha}_2 &= -0.7008 - 1.0107\,\mathrm{i}\,, \\ \boldsymbol{\beta}_0 &= 0.8405\,, & \boldsymbol{\beta}_1 &= -0.7890 + 1.0190\,\mathrm{i}\,, & \boldsymbol{\beta}_2 &= -0.7921 + 0.1183\,\mathrm{i}\,, \\ \mathbf{w}_0 &= 1.0\,, & \mathbf{w}_1 &= 0.2226 + 0.0030\,\mathrm{i}\,, & \mathbf{w}_2 &= 0.5319 - 0.4045\,\mathrm{i}\,, \end{aligned}$$

for the curve on the right with $(\eta, \rho) = (4.2000, 0.6682)$. Both curves satisfy (4). The values of the shape integrals are

$$L = 2.1610\,, \quad E = 15.806\,, \quad E_{\mathrm{rmf}} = 12.807$$

for the curve on the left, and

$$L = 1.9263\,, \quad E = 19.945\,, \quad E_{\mathrm{rmf}} = 15.998\,,$$

for the curve on the right. Figure 4 shows the variation of the ERF and RMF along the RRMF quintic on the left in Figure 3.

Example 3. In the final example, we consider data obtained by sampling the circular helix $\mathbf{r}(t) = (\sin(t), \cos(t), t)$ at ten equidistant points on $t \in [\,0, \frac{9}{2}\pi\,]$. The resulting G^1 RRMF quintic interpolants are illustrated in Figure 5.

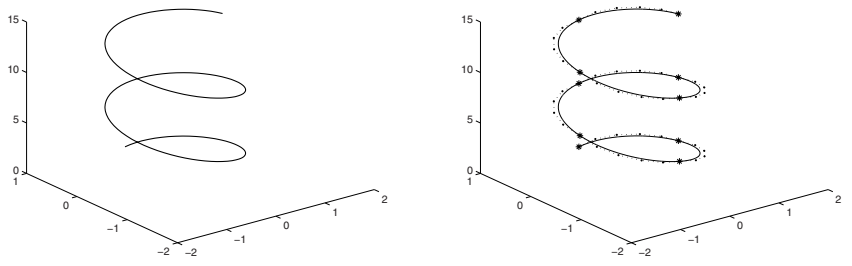


Fig. 5. A piecewise G^1 RRMF quintic interpolant (right) to Hermite data sampled from the circular helix (left). For each spline segment (delimited by the * symbols) the RRMF quintic Hermite interpolant is shown together with its Bézier control polygon.

6 Closure

A method for computing quintic RRMF (rational rotation–minimizing frame) curves that interpolate spatial G^1 Hermite data has been presented. Such curves are useful in applications such as motion control, animation, and swept surface constructions. The method involves one free angular parameter, that can strongly influence the curve shape. Numerical experiments show that, for many Hermite data sets, interpolants of good shape can be obtained by the method, but the formulation of an automatic and efficient procedure for their selection is an open problem. Further open problems concern the existence and multiplicity of the interpolants to arbitrary Hermite data, and the geometrical significance of the parameter η. Because they are highly non–linear, these problems are non–trivial: we hope to address them more carefully in future studies.

References

1. Bishop, R.L.: There is more than one way to frame a curve. American Mathematical Monthly 82, 246–251 (1975)
2. Choi, H.I., Han, C.Y.: Euler–Rodrigues frames on spatial Pythagorean-hodograph curves. Computer Aided Geometric Design 19, 603–620 (2002)
3. Choi, H.I., Lee, D.S., Moon, H.P.: Clifford algebra, spin representation, and rational parameterization of curves and surfaces. Advances in Computational Mathematics 17, 5–48 (2002)
4. Farouki, R.T.: Exact rotation-minimizing frames for spatial Pythagorean-hodograph curves. Graphical Models 64, 382–395 (2002)
5. Farouki, R.T.: Pythagorean–Hodograph Curves: Algebra and Geometry Inseparable. Springer, Berlin (2008)
6. Farouki, R.T., al–Kandari, M., Sakkalis, T.: Hermite interpolation by rotation-invariant spatial Pythagorean-hodograph curves. Advances in Computational Mathematics 17, 369–383 (2002)
7. Farouki, R.T., Giannelli, C.: Spatial camera orientation control by rotation–minimizing directed frames. Computer Animation and Virtual Worlds 20, 457–472 (2009)
8. Farouki, R.T., Giannelli, C., Manni, C., Sestini, A.: Identification of spatial PH quintic Hermite interpolants with near–optimal shape measures. Computer Aided Geometric Design 25, 274–297 (2008)

9. Farouki, R.T., Giannelli, C., Manni, C., Sestini, A.: Quintic space curves with rational rotation–minimizing frames. Computer Aided Geometric Design 26, 580–592 (2009)
10. Farouki, R.T., Giannelli, C., Sestini, A.: Helical polynomial curves and double Pythagorean hodographs I. Quaternion and Hopf map representation. Journal of Symbolic Computation 44, 161–179 (2009)
11. Farouki, R.T., Han, C.Y.: Rational approximation schemes for rotation–minimizing frames on Pythagorean–hodograph curves. Computer Aided Geometric Design 20, 435–454 (2003)
12. Guggenheimer, H.: Computing frames along a trajectory. Computer Aided Geometric Design 6, 77–78 (1989)
13. Han, C.Y.: Nonexistence of rational rotation–minimizing frames on cubic curves. Computer Aided Geometric Design 25, 298–304 (2008)
14. Jüttler, B., Mäurer, C.: Cubic Pythagorean hodograph spline curves and applications to sweep surface modeling. Computer Aided Design 31, 73–83 (1999)
15. Klok, F.: Two moving coordinate frames for sweeping along a 3D trajectory. Computer Aided Geometric Design 3, 217–229 (1986)
16. Sir, Z., Jüttler, B.: Spatial Pythagorean hodograph quintics and the approximation of pipe surfaces. In: Martin, R., Bez, H., Sabin, M. (eds.) Mathematics of Surfaces XI, pp. 364–380. Springer, Berlin (2005)
17. Wang, W., Joe, B.: Robust computation of the rotation minimizing frame for sweep surface modelling. Computer Aided Design 29, 379–391 (1997)
18. Wang, W., Jüttler, B., Zheng, D., Liu, Y.: Computation of rotation minimizing frames. ACM Transactions on Graphics 27(1), Article 2, 1–18 (2008)

Multiresolution Analysis for Minimal Energy C^r-Surfaces on Powell-Sabin Type Meshes

Miguel A. Fortes[1], Pedro González[1], Maria Moncayo[2], and Miguel Pasadas[1]

[1] Departamento de Matemática Aplicada, Universidad de Granada.
Campus de Fuentenueva s/n, 18071 Granada, Spain
{mafortes,prodelas,mpasadas}@ugr.es
[2] Departamento de Matemática Aplicada y Estadística, Universidad Politécnica de Cartagena (UPCT), Campus Muralla del Mar, 30202 Cartagena, Spain
maria.moncayo@upct.es

Abstract. This paper is intended to provide a multiresolution analysis (MRA) scheme to obtain a sequence of C^r-spline surfaces over a Powell-Sabin triangulation of a polygonal domain approximating a Lagrangian data set and minimizing a certain "energy functional". We define certain non separable scaling and wavelet functions in bidimensional domains, and we give the decomposition and reconstruction formulas in the framework of lifting schemes. Two important applications of the theory are given: In the first one we develop an algorithm for noise reduction of signals. The second one is related to the localization of the regions where the energy of a given function is mostly concentrated. Some numerical and graphical examples for different test functions and resolution levels are given.

1 Introduction

Multiresolution and subdivision schemes are successfully applied in a variety of engineering fields. Subdivision schemes are based on refinement rules which are applied on a starting set of discrete data to generate a new *denser* set. The procedure is recursively used and, under some assumptions, it converges to a certain limit function [7,10].

Subdivision schemes are related with multiresolution algorithms. More precisely, the multiresolution algorithms connect a finite sequence a_L, which represents sampling of weighted-averaged of a function at resolution level L, with its multi-scale representation

$$\{a_k, d_k, d_{k+1}, \ldots, d_{L-1}\}, \tag{1}$$

where, for $0 \leq k \leq L-1$, the sequence a_k denotes an approximation of a_L at resolution k, and, for $k \leq j \leq L-1$, d_j represents the intermediate details needed to recover a_{j+1} from a_j. In this way, it is possible to obtain $\{a_{L-1}, d_{L-1}\}$ from a_L (decomposition process) and, reciprocally, we can also obtain a_L from $\{a_{L-1}, d_{L-1}\}$, or equivalently, a_L from (1) (reconstruction process). Wavelet decompositions constitutes a relevant theory of linear multi-scale representations [6,15]. In fact, the construction of multiresolution schemes according to different practical contexts is a growing relevant research area [4,13]. The theoretical analysis depends on the underlying functional space used to formulate the problem at issue and, as usual, the difficulty level increases with the spatial dimension.

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 209–223, 2010.

In this work we introduce a multiresolution analysis to obtain C^r-spline surfaces which approximates a Lagrangian data set in a certain type of domain $D \subset \mathbb{R}^2$, and minimizes an "energy functional". The minimization spaces are spline spaces constructed from the Powell-Sabin subtriangulation associated to different uniform Δ^1-type triangulations of $\overline{D}$. More precisely, for each resolution level we will consider a uniform Δ^1-type triangulation in such a way that the approximating spline spaces associated to consecutive levels are nested. Recently, some authors have developed subdivision algorithms regarding energy properties (see e.g. [9]) and Powell-Sabin splines (see e.g. [19] and references therein). In the context of compressing surfaces related to hierarchical bases, it can be consulted [11], where $\mathscr{C}^1$-cubic splines defined on triangulated quadrangulations are used. In the present work we obtain the corresponding decomposition and reconstruction algorithms and, by the use of them, we will compute the detail coefficients relative to the multiresolution representation (1) in the framework of the minimal energy surfaces. Our approach allows us to localize the regions where the energy of a given function is mostly concentrated, what is a relevant fact for practical purposes. We will be also able to use the developed algorithms to describe a process to denoise a given signal, what constitutes a field of increasing interest in the last years.

The organization of the paper is as follows: In Section 2 we describe the basic background on multiresolution techniques, while in Section 3 we develop the basic concepts needed to describe the construction of the minimal energy surfaces and we state the problem we want to solve. In Section 4 we derive the decomposition and reconstruction procedures in the framework of the multiresolution analysis scheme, and we make use of two theorems given in [2] and [3] in order to show that as the number of Lagrangian data and resolution level increases, some kind of convergence is guaranteed in the context of the present work. Finally, in Section 5, two applications of the developed theory are analyzed: one concerning noise reduction and the other one related to the localization of the regions of maximum energy of a given function.

2 Background on Multiresolution

A multiresolution scheme constitutes a suitable framework for the representation of a data set a_L into an approximation a_k, $0 \leq k \leq L-1$, and different detail levels $\{d_k, d_{k+1}, \ldots, d_{L-1}\}$. In this context, a collection of *approximation spaces* $\{V_j\}_{j\geq 0}$ of a suitable functional space $\mathscr{H}$ such that

$$V_0 \subset V_1 \subset \cdots \subset V_j \subset \cdots \subset \mathscr{H}, \tag{2}$$

and $\bigcup_{j\geq 0} V_j$ is dense in $\mathscr{H}$, is considered [6, 15]. The *detail spaces* W_j (for $j \geq 0$) are defined by the identity $V_{j+1} = V_j \bigoplus W_j$. Hence, we have

$$V_L = V_{L-1} \oplus W_{L-1} = \cdots = V_j \oplus W_j \oplus \cdots \oplus W_{L-1} \qquad (j \leq L-1).$$

In the classical wavelet theory, the multiresolution analysis is developed in $\mathscr{H} = L^2(\mathbb{R})$, and the spaces V_j are generated by dilation and translation of one fixed function, called the *scaling* function, while the detail spaces are generated by the application of the same procedure to the so-called *wavelet* function. The multiresolution analysis in

$L^2(\mathbb{R}^2)$ is based on tensor products of these functions [6,15]. The classical construction involves Fourier transforms techniques to obtain the scaling and wavelet functions in terms of the low-pass and high-pass filters, respectively. However, there are many situations in which the multi-scale approach is well suited to formulate certain problems but the Fourier techniques are no longer available to obtain the multiresolution algorithms. In such cases these algorithms are given by means of the so-called lifting schemes, i.e., a completely spatial framework without the frequency analysis [12,18].

Lifting methods can be regarded as a reimplementation of the decomposition and reconstruction processes, which are classically given by means of filter bank algorithms defined by the so called quadrature mirror filters. These four filters make use of the operations given by downsampling, upsampling and convolutions. The lifting scheme proposes to split the input data in *old* and *new* components. The terms *old* and *new* are used according to the hierarchy defined by the subdivision procedure (see Figure 1). The new data is predicted from the old data by the use of a predictor P. The predictor needs not to be exact, so the details d_j are defined by the difference

$$d_j = a_{j+1}^{new} - P(a_{j+1}^{old}). \tag{3}$$

Conversely, in practice, the data a_{j+1}^{old} is equal to a_j up to a filtered (or lifted) version of the details, by means of the operator U:

$$a_j = a_{j+1}^{old} + U(d_j). \tag{4}$$

The case $P = 1$ and $U = 1/2$ corresponds to the Haar wavelet transform (see [12], p. 24). In fact, all the classical wavelet transforms can be reformulated by means of lifting.

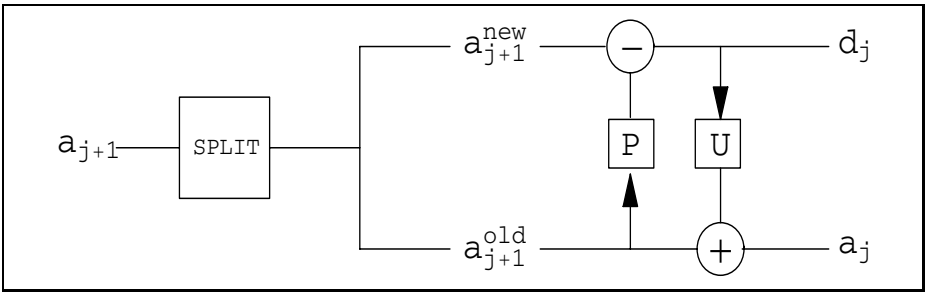


Fig. 1. General lifting scheme

The lifting transform is easily invertible. From (3) and (4) it follows

$$\begin{aligned} a_{j+1}^{new} &= d_j + P(a_j - U(d_j)) \\ a_{j+1}^{old} &= a_j - U(d_j) \end{aligned}.$$

So, the plus signs are replaced with minus signs and *split* is replaced with *merge*. The inversion involves the same operators, P and U. This fact constitutes a great advantage of lifting schemes in comparison with the classical inverse wavelet transforms. The lifting reconstruction diagram is:

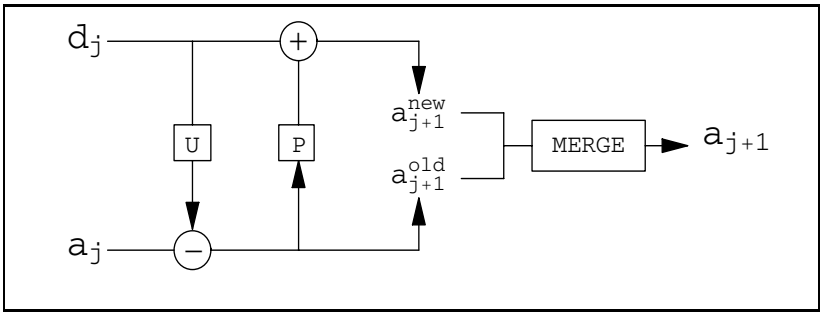


Fig. 2. The inverse lifting scheme

3 Background on Minimal Energy Surfaces

Let $\mathscr{T}_{\mathbb{R}^2}$ be any uniform Δ^1-type triangulation of the plane $\mathbb{R}^2$ induced by integer translates of $x=0, y=0$ and $x+y=0$, and let $D \subset \mathbb{R}^2$ be a polygonal domain such that $\overline{D}$ is the union of some triangles of $\mathscr{T}_{\mathbb{R}^2}$. Let us consider the Sobolev space $H^{r+1}(D)$ $(r \geq 1)$, whose elements are (classes of) functions u defined on D such that their partial derivatives (in the distribution sense) $\partial^\beta u$ belong to $L^2(D)$, with $\beta = (\beta_1, \beta_2) \in \mathbb{N}^2$ and $|\beta| = \beta_1 + \beta_2 \leq r+1$. In this space we consider the usual inner semi-products

$$(u,v)_t = \sum_{|\beta|=t} \int_D \partial^\beta u(x) \partial^\beta v(x) dx \quad \text{for} \quad t = 0, \ldots, r+1,$$

being $\partial^0 u(x) = u(x)$ for all $x \in D$, the corresponding semi-norms

$$|u|_t = (u,u)_t^{1/2} = \left(\sum_{|\beta|=t} \int_D \partial^\beta u(x)^2 dx \right)^{1/2} \quad \text{for} \quad t = 0, \ldots, r+1,$$

and the norm

$$\|u\|_{r+1} = \left(\sum_{t=0}^{r+1} |u|_t^2 \right)^{1/2}.$$

We will consider a uniform Δ^1-type triangulation $\mathscr{T}$ of $\overline{D}$ (see e.g. [5]), and the associated Powell-Sabin subtriangulation $\mathscr{T}^*$ of $\mathscr{T}$ (see [14] or [16]). Such subtriangulations are obtained by subdividing each triangle $T \in \mathscr{T}$ into six subtriangles by connecting each vertex of T to the midpoint of the opposite side. Hence, all these micro triangles have the barycenter of T as a common vertex. Nevertheless, Powell-Sabin subtriangulation can be also obtained by using the incenter of T instead of the barycenter in the split procedure for a larger class of triangulations (see [17]).

Let $n = n(r) = 2r+1$ for r even and $n = n(r) = 2r$ for r odd. Let $[x]$ denote the integer part of x. We consider the set

$$\mathscr{S}_n^{r,r+[\frac{r}{2}]}(\overline{D}, \mathscr{T}) = \{v \in C^r(D) : v|_T \in \mathscr{S}_n^{[\frac{n-1}{2}]+1, r+[\frac{r}{2}]}(T, \mathscr{T}^*), \forall T \in \mathscr{T}\},$$

where

$$\mathscr{S}_n^{[\frac{n-1}{2}]+1,r+[\frac{r}{2}]}(T,\mathscr{T}^*)=\left\{v\in C^{[\frac{n-1}{2}]+1}(T):\begin{array}{l} v|_{T'}\in\mathbb{P}_n(T'),\forall T'\in\mathscr{T}^*,T'\subset T,\\ \text{and } v \text{ is of class } C^{r+[\frac{r}{2}]}\\ \text{at the vertices of } T\end{array}\right\},$$

and $\mathbb{P}_n(T')$ indicates the space of bivariate polynomials of total degree at most n over T'. In [14] it is shown that given a function $f\in C^m(\overline{D}), m\geq r+[\frac{r}{2}]$, there exists a unique function $v\in\mathscr{S}_n^{r,r+[\frac{r}{2}]}(\overline{D},\mathscr{T})$ such that the values of v and all its partial derivatives up to order $r+[\frac{r}{2}]$ coincide with those of f at all the vertices of $\mathscr{T}$.

Let us consider a finite subset $\mathscr{P}=\{p_1\ldots,p_q\}$ of points in $\overline{D}$ and a given vector of real values $Z=(z_i)_{i=1}^q\in\mathbb{R}^q$. From the continuous injection of $H^{r+1}(D)$ into $C^0(\overline{D})$, we can define the evaluation operator $\rho(v):=(v(p_i))_{i=1}^q\in\mathbb{R}^q$ for all $v\in H^{r+1}(D)$.

We are looking for a C^r-surface, $r\geq 1$, that approximates the points $\{(p_i,z_i)\}_{i=1}^q\subset\mathbb{R}^3$ by minimizing the functional energy

$$J(v)=<\rho(v)-Z>_q^2+\sum_{t=1}^{r+1}\tau_t|v|_t^2,$$

where $<\cdot>_q$ represents the usual Euclidean norm in $\mathbb{R}^q$, $\tau:=(\tau_1,\ldots,\tau_{r+1})$, $\tau_i\in[0,+\infty)$ for all $i=1,\ldots,r$ and $\tau_{r+1}\in(0,+\infty)$. Observe that the first term of J measures (in the least squares sense) how well v approximates the values in Z, while the second one represents the "minimal energy condition" over the semi-norms $|\cdot|_1,\ldots,|\cdot|_{r+1}$ weighted by the parameters $\tau_1,\ldots,\tau_{r+1}$, respectively.

More precisely, for each resolution level, the minimization problem under consideration is:

Problem 1. Find an element $\sigma\in\mathscr{S}_n^{r,r+[\frac{r}{2}]}(\overline{D},\mathscr{T})$ such that $J(\sigma)\leq J(v)$ for all $v\in\mathscr{S}_n^{r,r+[\frac{r}{2}]}(\overline{D},\mathscr{T})$.

In [2] it is shown that, under the condition

$$Ker(\rho)\cap\mathbb{P}_r(D)=\{0\},$$

Problem 1 has a unique solution (the main idea of the proof consists of applying the Lax-Milgram Lemma to the bilinear form $a(u,v)=<\rho(u),\rho(v)>_q+\sum_{t=1}^{r+1}\tau_t(u,v)_t$, where $<\cdot,\cdot>_q$ represents the usual Euclidean inner product in $\mathbb{R}^q$, and to the linear form $\varphi(v)=<Z,\rho(v)>_q$, for $u,v\in\mathscr{S}_n^{r,r+[\frac{r}{2}]}(\overline{D},\mathscr{T})$).

Moreover, if we let $N=\dim(\mathscr{S}_n^{r,r+[\frac{r}{2}]}(\overline{D},\mathscr{T}))$, we consider a basis $\{v_1,\ldots,v_N\}$ of the finite element space $\mathscr{S}_n^{r,r+[\frac{r}{2}]}(\overline{D},\mathscr{T})$, and we denote the unique solution of Problem 1 as $\sigma=\sum_{i=1}^N\alpha_iv_i$, then the vector of coefficients $X=\left((\alpha_i)_{i=1}^N\right)^t$ is obtained as the unique solution of the linear system

$$CX=T, \tag{5}$$

where

$$C=(a(v_i,v_j))_{i,j=1}^N \text{ and } T=\left((\varphi(v_i))_{i=1}^N\right)^t.$$

Clearly, the coefficient matrix C of (5) is symmetric and, in [2], it is also shown that C is positive definite.

4 Multiresolution Analysis

In the present section we will use the following notation: Given two sets of functions $\mathscr{F} = \{f_i\}_{i=1}^{d_f}$ and $\mathscr{G} = \{g_j\}_{j=1}^{d_g}$ of the space $H^{r+1}(D)$ we will denote

$$C_{\mathscr{F},\mathscr{G}} = (a(f_i, g_j))_{\substack{i=1,\ldots,d_f \\ j=1,\ldots,d_g}}$$

and

$$T_{\mathscr{F}} = \left((\varphi(f_i))_{i=1}^{d_f}\right)^t$$

(observe that the bilinear form a and the linear form φ considered previously can be extended to $H^{r+1}(D)$).

Let $\{\mathscr{T}_j\}_{j\geq 0}$ be a sequence of refined uniform Δ^1-triangulations of $\overline{D}$ in such a way that the sequence of approximation spaces $V_j = S_n^{r,r+[\frac{r}{2}]}(\overline{D}, \mathscr{T}_j)$ is nested as in (2), and let us denote $n_j = \dim(V_j)$. From Example 7.2.7. of [1] we know that $V_j \subset H^{r+1}(D)$. On the other hand, given $u \in H^{r+1}(D)$ and $\varepsilon > 0$, by applying the Stone-Weierstrass theorem and the continuous injection of $H^{r+1}(D)$ into $C^0(\overline{D})$, there exists $p \in \bigcup_{n\geq 0} \mathbb{P}_n(D)$ and $C > 0$ such that

$$\|u - p\|_{r+1} \leq C\|u - p\|_\infty < \frac{\varepsilon}{2}.$$

By applying Lemma 6 of [2], we have that for h small enough there exists $v \in V_{j(h)}$, where $j(h) \geq 0$ is such that the diameter of the triangles of $\mathscr{T}_{j(h)}$ is h, such that

$$\|p - v\|_{r+1} < \frac{\varepsilon}{2},$$

and, therefore,

$$\|u - v\|_{r+1} < \varepsilon.$$

Hence $\bigcup_{j\geq 0} V_j$ is dense in $H^{r+1}(D)$ with respect to $\|\cdot\|_{r+1}$ and, in particular, the approximation spaces V_j constitute a MRA in $\mathscr{H} = H^{r+1}(D)$:

$$V_0 \subset V_1 \subset \cdots \subset V_j \subset \cdots \subset H^{r+1}(D).$$

Let $\mathscr{B}_0 = \{\varphi_0^k\}_{k=1}^{n_0}$ be the usual Hermite basis of V_0 and, for all $j \geq 0$, let us consider $\Psi_j = \{\psi_j^k\}_{k=1}^{n_{j+1}-n_j}$ the functions of the Hermite basis of V_{j+1} associated to the knots belonging to $\mathscr{T}_{j+1}$ that do not belong to $\mathscr{T}_j$. Then, it is clear that if we define

$$\begin{cases} \varphi_{j+1}^k = \varphi_j^k & \text{for } k = 1, \ldots, n_j \\ \varphi_{j+1}^k = \psi_j^{k-n_j} & \text{for } k = n_j + 1, \ldots, n_{j+1}, \end{cases}$$

we have that $\mathscr{B}_{j+1} = \{\varphi_{j+1}^k\}_{k=1}^{n_{j+1}} = \mathscr{B}_j \cup \Psi_j$ is a basis with local support of the space V_{j+1} for $j \geq 0$.

Let σ_j be the unique solution of Problem 1 for $\mathscr{T} = \mathscr{T}_j$. Then, there exists a sequence $a_j = \{a_j^k\}_{k=1}^{n_j}$ such that

$$\sigma_j = \sum_{k=1}^{n_j} a_j^k \varphi_j^k, \tag{6}$$

and, from (5), it follows that, for all $j \geq 0$,

$$C_{\mathscr{B}_j,\mathscr{B}_j} a_j = T_{\mathscr{B}_j}. \tag{7}$$

Moreover, for all $j \geq 1$ we have that

$$C_{\mathscr{B}_j,\mathscr{B}_j} = \left(\begin{array}{c|c} C_{\mathscr{B}_{j-1},\mathscr{B}_{j-1}} & C_{\mathscr{B}_{j-1},\Psi_{j-1}} \\ \hline C_{\Psi_{j-1},\mathscr{B}_{j-1}} & C_{\Psi_{j-1},\Psi_{j-1}} \end{array}\right) \tag{8}$$

and

$$T_{\mathscr{B}_j} = \left(\begin{array}{c} T_{\mathscr{B}_{j-1}} \\ \hline T_{\Psi_{j-1}} \end{array}\right). \tag{9}$$

Hence, if we denote the element $\sigma_j = \sum_{k=1}^{n_j} a_j^k \varphi_j^k$ as

$$\sum_{k=1}^{n_{j-1}} \widetilde{a}_{j-1}^k \varphi_{j-1}^k + \sum_{k=1}^{n_j - n_{j-1}} d_{j-1}^k \psi_{j-1}^k,$$

that is,

$$\{a_j^1, a_j^2, \ldots, a_j^{n_j}\} = \{\widetilde{a}_{j-1}^1, \widetilde{a}_{j-1}^2, \ldots, \widetilde{a}_{j-1}^{n_{j-1}}, d_{j-1}^1, d_{j-1}^2, \ldots, d_{j-1}^{n_j - n_{j-1}}\}, \tag{10}$$

then from (7), (8), (9) and (10), it holds

$$C_{\mathscr{B}_{j-1},\mathscr{B}_{j-1}} \widetilde{a}_{j-1} + C_{\mathscr{B}_{j-1},\Psi_{j-1}} d_{j-1} = T_{\mathscr{B}_{j-1}} \tag{11}$$

$$C_{\Psi_{j-1},\mathscr{B}_{j-1}} \widetilde{a}_{j-1} + C_{\Psi_{j-1},\Psi_{j-1}} d_{j-1} = T_{\Psi_{j-1}}, \tag{12}$$

where $\widetilde{a}_{j-1} = \left((\widetilde{a}_{j-1}^k)_{k=1}^{n_{j-1}}\right)^t$ and $d_{j-1} = \left((d_{j-1}^k)_{k=1}^{n_j - n_{j-1}}\right)^t$.

By replacing $T_{\mathscr{B}_{j-1}}$ by $C_{\mathscr{B}_{j-1}\mathscr{B}_{j-1}} a_{j-1}$ in (11) we obtain

$$a_{j-1} - \widetilde{a}_{j-1} = (C_{\mathscr{B}_{j-1},\mathscr{B}_{j-1}})^{-1} C_{\mathscr{B}_{j-1},\Psi_{j-1}} d_{j-1}.$$

From this last equation and by the use of the notation introduced in (6), the decomposition algorithm can be formulated as follows:

$$\begin{cases} d_{j-1} = \left((a_j^k)_{k=n_{j-1}+1}^{n_j}\right)^t \\ a_{j-1} = \widetilde{a}_{j-1} + (C_{\mathscr{B}_{j-1},\mathscr{B}_{j-1}})^{-1} C_{\mathscr{B}_{j-1},\Psi_{j-1}} d_{j-1}, \end{cases} \tag{13}$$

and the reconstruction algorithm is:

$$\begin{cases} \left((a_j^k)_{k=n_{j-1}+1}^{n_j}\right)^t = d_{j-1} \\ \left((a_j^k)_{k=1}^{n_{j-1}}\right)^t = \widetilde{a}_{j-1} = a_{j-1} - (C_{\mathscr{B}_{j-1},\mathscr{B}_{j-1}})^{-1} C_{\mathscr{B}_{j-1},\Psi_{j-1}} d_{j-1}. \end{cases} \tag{14}$$

Observe that by taking P and U of expressions (3) and (4) as the null operator and $C^{-1}_{\mathscr{B}_j,\mathscr{B}_j} C_{\mathscr{B}_j\Psi_j}$ respectively, the previous algorithms are formulated in the framework of the lifting scheme recalled in Section 2.

In [2] and [3] two convergence results (for exact and noisy data, respectively) are shown in the context of the present work. More precisely, for each $s \in \mathbb{N}^*$, let $\mathscr{P}^s \subset \overline{D}$ be a finite set consisting of $q = q(s)$ points and $Z^s \in \mathbb{R}^q$ be a vector of real values. Let $d = d^s = \sup_{x \in D}\{\delta(x, \mathscr{P}^s)\}$, where δ denotes the Euclidean distance in $\mathbb{R}^2$. Let $g \in C^{n+1}(D)$ and h_j be the diameter of the triangles of $\mathscr{T}_j$ for $j \geq 0$. Let us suppose that $\{h_j\}_{j \geq 0} \subset \mathbb{R}_*^+$ admits 0 as an accumulation point. Let $\sigma_{s,j}$ be the solution of Problem 1 for $\mathscr{T} = \mathscr{T}_j$, $\mathscr{P} = \mathscr{P}^s$ and $Z = Z^s$. Then, in [2] it is shown that:

Theorem 2. *Under the hypotheses*

$$d = O\left(\frac{1}{s}\right), \quad s \to +\infty, \tag{15}$$

$$\exists\, C > 0 \text{ and } \widetilde{s} \in \mathbb{N}^* \text{ such that } q(s) \leq Cs^2,\ \forall s \geq \widetilde{s}, \tag{16}$$

$$\tau_{r+1} = o(s^2), \quad s \to +\infty, \tag{17}$$

$$\tau_m = o(\tau_{r+1}), \quad s \to +\infty, \quad \forall m = 1, \ldots, r, \tag{18}$$

$$\frac{s^2 h_j^{2n+2}}{\tau_{r+1}} = o(1), \quad s \to +\infty, \tag{19}$$

it holds that $\lim_{s \to +\infty} \|g - \sigma_{s,j}\|_{r+1} = 0$.

On the other hand, let us consider for each $s \in \mathbb{N}^*$ a set $\widetilde{Z}^s = Z^s + \vartheta^s$ of noisy data points, where $\vartheta^s = (\vartheta_i^s)_{i=1}^q$ is a error vector in $\mathbb{R}^q$ of white noise type for all $s \in \mathbb{N}^*$, and let $\widetilde{\sigma}_{s,j}$ be the solution of Problem 1 for $\mathscr{T} = \mathscr{T}_j$, $\mathscr{P} = \mathscr{P}^s$ and $Z = \widetilde{Z}^s$. Then, in [3] it is shown that:

Theorem 3. *If in addition to (15), (16) and (18), the following conditions hold:*

$$d = O\left(\inf_{a,b \in \mathscr{P}^s, a \neq b} < a - b >_2\right), \quad s \to +\infty, \tag{20}$$

$$\exists C > 0,\ \exists\, \alpha \in (1,2) \text{ such that } \lim_{s \to +\infty} \frac{\tau_{r+1}}{s^\alpha} = C, \tag{21}$$

$$s^{2-\alpha} h_j^2 \to +\infty, \quad s \to +\infty, \tag{22}$$

$$s^{2-\alpha} h_j^{2n+2} \to 0, \quad s \to +\infty, \tag{23}$$

then $\lim_{s \to +\infty} \|g - \widetilde{\sigma}_{s,j}\|_{r+1} = 0$ *almost surely.*

We also want to mention that, if $D = [a,b] \times [a,b]$ is a square domain, then (19), which is the unique hypothesis involving the sequence of triangulations, could be replaced by a new condition (19′) in which the resolution level j appears. More precisely, in this case it is easy to compute that $h_j = \frac{1}{2^j}\gamma$, where $\gamma = \sqrt{2}(b-a)$, and hence, Equation (19) can be replaced by

$$\frac{s^2}{\tau_{r+1}(4^{n+1})^j} = o(1), \quad s \to +\infty. \tag{19′}$$

Analogously, in the convergence result for noisy data, conditions (22) and (23) can be stated in terms of the resolution level j:

$$\frac{s^{2-\alpha}}{4^j} \to +\infty, \quad s \to +\infty, \tag{22'}$$

$$\frac{s^{2-\alpha}}{(4^{n+1})^j} \to +\infty, \quad s \to +\infty. \tag{23'}$$

5 Applications

In order to show the effectiveness of the given algorithms, we will consider two applications. In the first one, the noise of a surface is reduced and in the second one, the subdomain in which the energy of a surface is maximally concentrated is spatially localized.

In both cases, in order to establish the multiresolution scheme required to formulate the algorithms, we have considered the domain $D = (0,1) \times (0,1)$ and triangulations $\mathscr{T}_0, \mathscr{T}_1$ and $\mathscr{T}_2$ (which are represented in Figure 3).

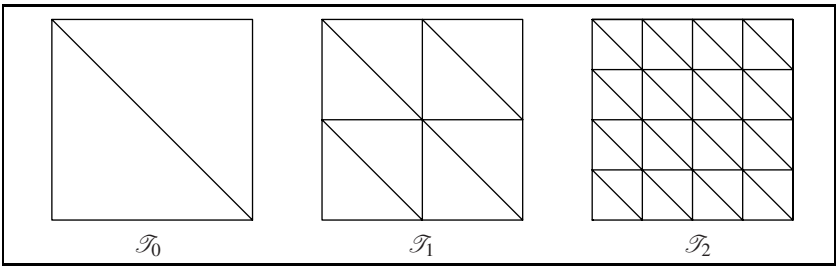


Fig. 3. Triangulations $\mathscr{T}_0, \mathscr{T}_1$ and $\mathscr{T}_2$ of $D = [0,1] \times [0,1]$

Both applications are developed with C^1-quadratic splines, that is, $V_j = \mathscr{S}_2^{1,1}(D, \mathscr{T}_j)$, and therefore $n_j = \dim(V_j) = 3(2^j+1)^2$. To obtain the basis functions $\mathscr{B}_j$ of the vector space V_j, we have considered the following three steps:

1. Let us consider the reference triangle T_0 with vertices $A_1 = (0,1), A_2 = (0,0)$ and $A_3 = (1,0)$, and the linear functionals $L_i(f) = f(A_i)$, $L_{i+3}(f) = \frac{\partial f}{\partial x}(A_i)$ and $L_{i+6}(f) = \frac{\partial f}{\partial y}(A_i)$ for $i = 1,2,3$. Let us consider then a basis of functions $\{w_1, \ldots, w_9\}$ over T_0 that verify $L_i(w_j) = \delta_{ij}$ for $i, j = 1, \ldots, 9$. Let $\{T_1, \ldots, T_6\}$ be the microtriangles of the Powell-Sabin triangulation of T_0:

   Over each triangle T_d, every polynomial p of total degree two can be expressed as $p(x) = \sum_{\substack{i,j,k=0,1,2 \\ i+j+k=2}} c_{ijk}^d \lambda_1^i \lambda_2^j \lambda_3^k$, where $(\lambda_1, \lambda_2, \lambda_3)$ is the barycentric coordinate vector of x with respect to T_d, for all $x \in T_d$. By applying the relations that must verify the B-coefficients of a given function f in order to be of class C^1 (see [8]), we determine the B-coefficients of the basis functions $\{w_i\}_{i=1}^9$.

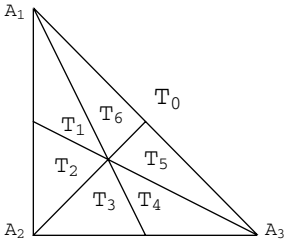


Fig. 4. Powell-Sabin triangulation of T_0

2. By composing the affine transformation that carries any arbitrary right triangle $T \in \mathscr{T}_j$ into the reference triangle T_0 with the functions $\{w_i\}_{i=1}^{9}$ described above, we obtain the usual Hermite functions basis (up to constants) over T.
3. Finally, by properly joining the functions above we get the basis $\mathscr{B}_0$, and in general, for $j \geq 0$, the basis $\mathscr{B}_{j+1}$ is obtained by joining the basis $\mathscr{B}_j$ with the usual Hermite functions of V_{j+1} associated to the $2^{j+1}(1+3\cdot 2^{j-1})$ knots belonging to $\mathscr{T}_{j+1}$ that do not belong to $\mathscr{T}_j$.

5.1 Noise Reduction

We have considered a set $\mathscr{P} = \{p_i\}_{i=1}^{2000}$ of points arbitrarily distributed over the domain D, and a vector of values $Z = \{f(p_i) + \vartheta_i\}_{i=1}^{2000}$, where f is the test function represented in Figure 5 and defined by

$$f(x,y) = e^{8\left(-(x-0.5)^2-(y-0.5)^2\right)},$$

and $\vartheta = \{\vartheta_i\}_{i=1}^{2000}$ is a noisy vector with zero mean and deviation $\mu = 5\cdot 10^{-2}$. The parameters values used in the example are $\tau_1 = 10^{-3}$ and $\tau_2 = 10^{-5}$.

The process used to reduce the noise is as follows:

Step 1. We have considered the resolution level $L = 3$. By substituting in expression (6) the coefficients a_3 obtained with Equation (7), we get the approximation spline $\sigma_3 = \sum_{k=1}^{n_3} a_3^k \varphi_3^k$ (see Figure 6).

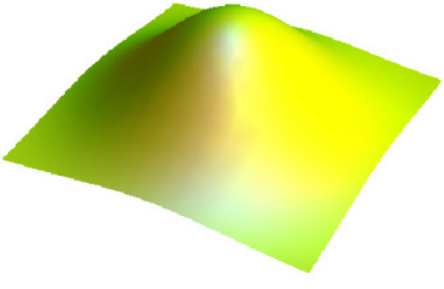


Fig. 5. Test function f

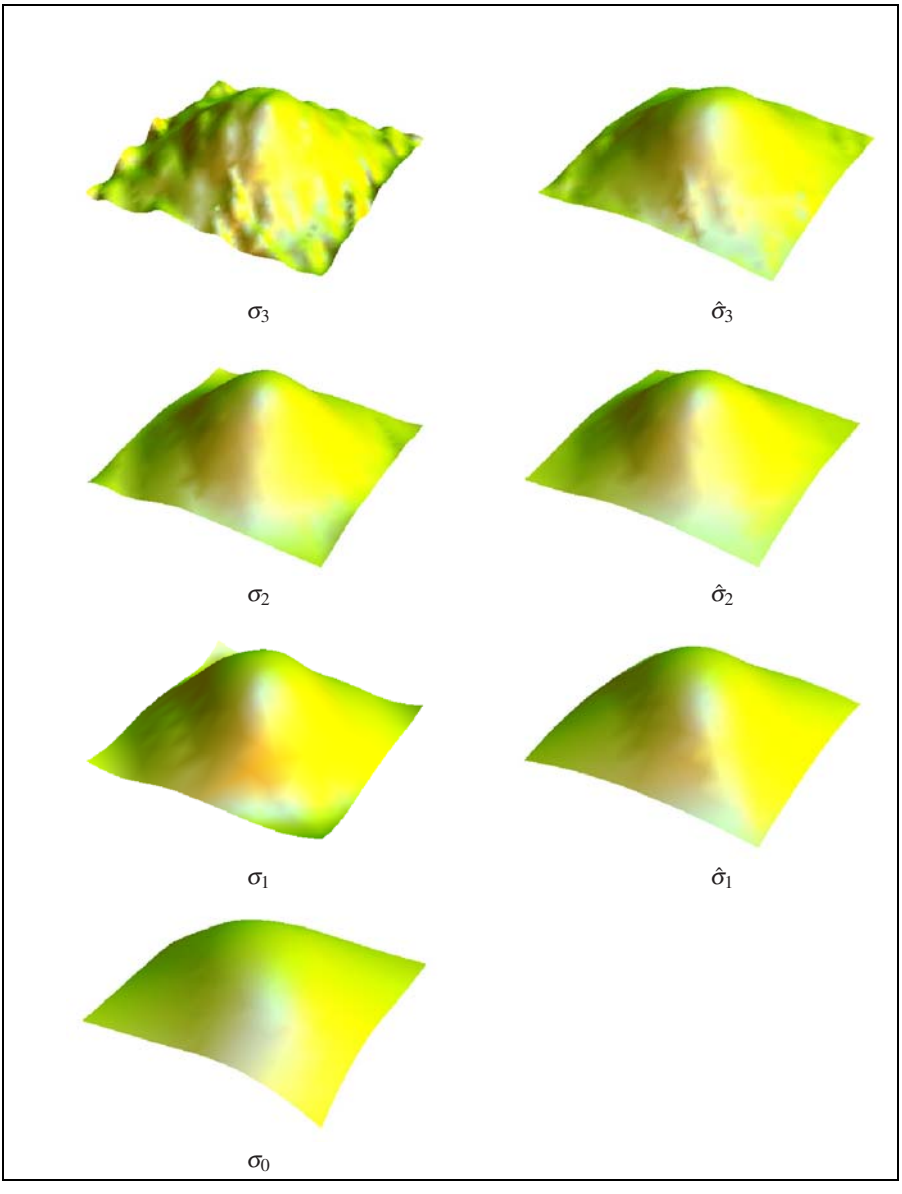


Fig. 6. Approximating splines and perturbed splines for different resolution levels

Step 2. By applying recursively the decomposition algorithm (13) we obtain the approximation coefficients a_t and the detail coefficients d_t from a_3 for $t = 2,1,0$ (in Figure 6 are represented the approximating splines $\sigma_t = \sum_{k=1}^{n_t} a_t^k \varphi_t^k$ for $t = 0,1,2,3$).

Step 3. In order to reduce the noise we have considered the following threshold method (used before in [19]) to define the perturbed detail coefficients $\hat{d}_t = \{\hat{d}_t^k\}_{k=1}^{n_{t+1}-n_t}$:

$$\hat{d}_t^k = \begin{cases} d_t^k & \text{if } |d_t^k| > 0.1 \max\{|d_t^k|\}_{k=1}^{n_{t+1}-n_t} \\ 0 & \text{if } |d_t^k| \leq 0.1 \max\{|d_t^k|\}_{k=1}^{n_{t+1}-n_t} \end{cases}$$

for $t = 0,1,2$ and $k = 1,\ldots,n_{t+1} - n_t$.

Step 4. With the reconstruction algorithm (14) we recursively obtain, from $\hat{a}_t$ and $\hat{d}_t$, the perturbed approximation coefficients $\hat{a}_{t+1} = \{\hat{a}_{t+1}^k\}_{k=1}^{n_{t+1}}$ for $t = 0,1,2$ (we define $\hat{a}_0 = a_0$).

Step 5. Finally, the perturbed approximation splines, in which the noise is reduced, are

$$\hat{\sigma}_t = \sum_{k=1}^{n_t} \hat{a}_t^k \varphi_t^k \qquad \text{for } t = 0,1,2,3.$$

In Figure 6 the splines $\hat{\sigma}_t$, for $t = 1,2,3$, are represented.

5.2 Localization of Energy

In this second application we propose to detect the regions in which the energy of a given function is maximally concentrated. We have considered the very localized test function defined by

$$g(x,y) = \begin{cases} 1 - 8\sqrt{(x-a)^2 + (y-a)^2} & \text{if } (x-a)^2 + (y-a)^2 \leq \dfrac{1}{8^2} \\ 1 - 8\sqrt{(x-b)^2 + (y-b)^2} & \text{if } (x-b)^2 + (y-b)^2 \leq \dfrac{1}{8^2} \\ 0 & \text{otherwise} \end{cases},$$

where $a = \dfrac{1}{4} + \dfrac{\pi}{100}$ and $b = \dfrac{3}{4} - \dfrac{\pi}{100}$. In Figure 7 we can see that the energy of g is concentrated around the two vertices.

The procedure starts as in Application 1:

Step 1. We have considered the resolution level $L = 3$. By substituting in expression (6) the coefficients a_3 obtained with Equation (7), we get the approximation function σ_3.

Step 2. By applying recursively the decomposition algorithm (13) we obtain, from a_3, the approximation coefficients a_t and the detail coefficients d_t for $t = 2,1,0$.

Step 3. For each resolution level $t = 0,1,2$, we have partitioned the set of the detail coefficients $d_t = \{d_t^k\}_{k=1}^{n_{t+1}-n_t}$ into five blocks $\mathscr{D}_t^k$ for $k = 0,\ldots,4$ in the following way:

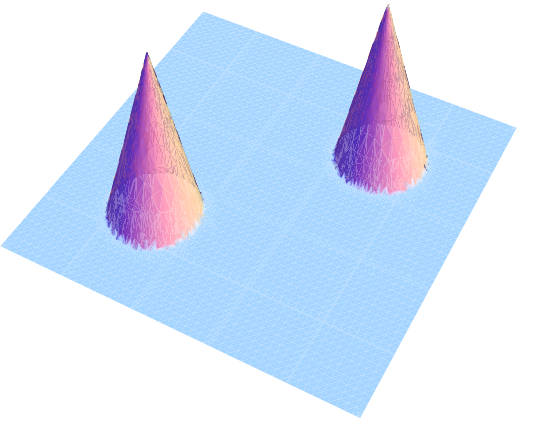


Fig. 7. Test function g

Let

$$\begin{aligned} M_t &= \max\{|d_t^k|\}_{k=1}^{n_{t+1}-n_t}, \\ m_t &= \min\{|d_t^k|\}_{k=1}^{n_{t+1}-n_t} \text{ and} \\ h_t &= M_t - m_t. \end{aligned}$$

Then

$$d_t^k \in \begin{cases} \mathscr{D}_t^0 & \text{if } |d_t^k| \in \left[m_t, m_t + \frac{4}{11}h_t\right] \\ \mathscr{D}_t^1 & \text{if } |d_t^k| \in \left(m_t + \frac{4}{11}h_t, m_t + \frac{5}{11}h_t\right] \\ \mathscr{D}_t^2 & \text{if } |d_t^k| \in \left(m_t + \frac{5}{11}h_t, m_t + \frac{6}{11}h_t\right] \\ \mathscr{D}_t^3 & \text{if } |d_t^k| \in \left(m_t + \frac{6}{11}h_t, m_t + \frac{7}{11}h_t\right] \\ \mathscr{D}_t^4 & \text{if } |d_t^k| \in \left(m_t + \frac{7}{11}h_t, M_t\right] \end{cases} \tag{24}$$

Step 4. To each triangle $T \in \mathscr{T}_t$ $(t = 1,2,3)$ we have assigned a number n_T defined as follows:

$$n_T = \max\{m \in \{0,1,2,3,4\} : d_t^k \in \mathscr{D}_t^m \text{ for all } k \text{ such that } T \subset supp(\psi_t^k)\}.$$

Note that the largest detail coefficients belong to $\mathscr{D}_t^4$. This means that this block is related to the regions around the vertices.

Step 5. Finally, in order to emphasize the energy distribution of g according to the contour plot map of g given in Figure 8, we have considered, for each resolution level $t = 1,2,3$, a graphic representation of $\mathscr{T}_t$ in which the triangles $T \in \mathscr{T}_t$ have been colored by the following criteria: black if $n_T = 0$, dark gray if $n_T = 1$, medium gray if $n_T = 2$, light gray if $n_T = 3$ and white if $n_T = 4$. Such representations are given in Figure 8.

A comparison between the different graphics of Figure 8 makes possible to conclude that, as the resolution level increases, the regions in white tend to the regions in which the energy of g is maximally concentrated.

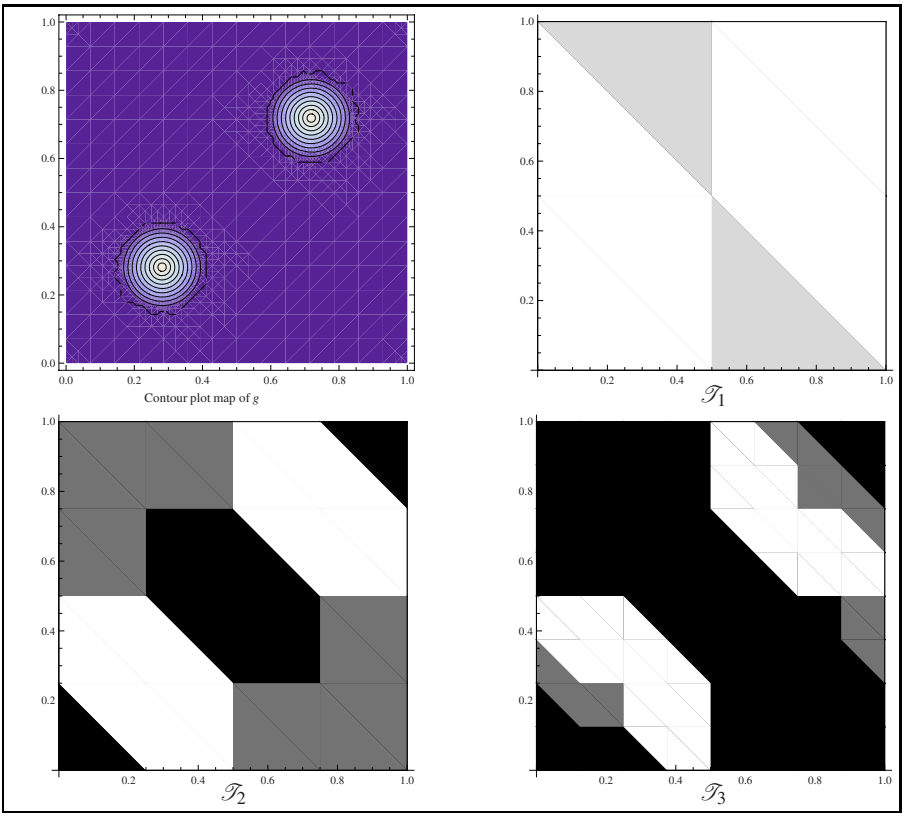


Fig. 8. Map of the energy of g for different resolutions levels

References

1. Atkinson, K., Han, W.: Theoretical Numerical Analysis, 2nd edn. Springer, Heidelberg (2005)
2. Barrera, D., Fortes, M.A., González, P., Pasadas, M.: Minimal energy C^r- surfaces on uniform Powell-Sabin type meshes. Estimation of the smoothing parameters. Mathematics and Computers in Simulation 77, 161–169 (2008)
3. Barrera, D., Fortes, M.A., González, P., Pasadas, M.: Minimal energy C^r-surfaces on uniform Powell-Sabin type meshes for noisy data. Journal of Computational and Applied Mathematics 218(2), 592–602 (2008)
4. Chui, C.K., Wang, J.Z.: Wavelet-based minimal-energy approach to image restoration. Appl. Comput. Harmon. Anal. 23, 114–130 (2007)
5. Davydov, O., Nürnberger, G., Zeilfelder, F.: Approximation order of bivariate spline interpolation for arbitrary smoothness. J. Comput. Appl. Math. 90, 117–134 (1998)
6. Daubechies, I.: Ten Lectures on Wavelets. SIAM, Philadelphia (1992)
7. Dyn, N.: Interpolatory Subdivision Schemes. In: Iske, A., Quak, E., Floater, M.S. (eds.) Tutorials on Multiresolution in Geometric Modelling, pp. 25–45. Springer, Heidelberg (2002)

8. Farin, G.: Bézier polynomials over triangles and the construction of piecewise C^r-polynomials. Technical Report TR/91, Brunel University, Uxbridge, England (1980)
9. Ginkel, I., Umlauf, G.: Local energy-optimizing subdivision algorithms. Computer Aided Geometric Design 25, 137–147 (2008)
10. Harten, A.: Multiresolution representation of data II. SIAM J. Numer. Anal. 33(3), 1205–1256 (1996)
11. Hong, D., Schumaker, L.L.: Surface compression using a space of $\mathscr{C}^1$ cubic splines with a hierarchical basis. Computing 72, 79–92 (2004)
12. Jansen, M., Oonincx, P.: Second Generation Wavelets and Applications. Springer, London (2005)
13. Jouini, A., Kratou, M.: Wavelet bases on a manifold. J. Funct. Anal. 248, 128–151 (2007)
14. Laghchim-Lahlou, M., Sablonnière, P.: C^r-finite elements of Powell-Sabin type on the three direction mesh. Adv. in Comput. Math. 6, 191–206 (1996)
15. Mallat, S.: A Wavelet Tour of Signal Processing. Academic Press, London (1999)
16. Powell, M.J.D., Sabin, M.A.: Piecewise quadratic approximations on triangles. ACM Trans. Math. Software 3(4), 316–325 (1977)
17. Sablonnière, P.: Error Bounds for Hermite Interpolation by Quadratic Splines on an α-triangulation. IMA Journal of Numerical Analysis 7, 495–508 (1987)
18. Sweldens, W.: The lifting scheme: A construction of second generation wavelets. SIAM J. Math. Anal. 29(2), 551–546 (1998)
19. Windmolders, J., Vanraes, E., Dierckx, P., Bultheel, A.: Uniform Powell Sabin splines wavelets. Journal of Computational and Applied Mathematics 154, 125–142 (2003)

Segmentation of 3D Tubular Structures by a PDE-Based Anisotropic Diffusion Model

Elena Franchini, Serena Morigi, and Fiorella Sgallari

Department of Mathematics-CIRAM, University of Bologna, Bologna, Italy
{franchini,morigi,sgallari}@dm.unibo.it

Abstract. Many different approaches have been proposed for segmenting vessels, or more generally tubular-like structures from 2D/3D images. In this work we propose to reconstruct the boundaries of 2D/3D tubular structures by continuously deforming an initial distance function following the Partial Differential Equation (PDE)-based diffusion model derived from a minimal volume-like variational formulation. The gradient flow for this functional leads to a non-linear curvature motion model. An anisotropic variant is provided which includes a diffusion tensor aimed to follow the tube geometry. Space discretization of the PDE model is obtained by finite volume schemes and semi-implicit approach is used in time/scale. The use of an efficient strategy to apply the linear system iterative solver allows us to reduce significantly the numerical effort by limiting the computation near the structure boundaries. We illustrate how the proposed method works to segment 2D/3D images of synthetic and medical real data representing branching tubular structures.

1 Introduction

This paper considers the segmentation of branching tubular objects from 2D/3D images. This problem arises in several application fields, like road map extraction in aerial photography and in medical image analysis where the interest focuses on the extraction of anatomical surfaces of tubular structures, like blood vessels. Indeed, problems like aneurysm or stenosis can occur in a vessel, and the clinicians need tools both for computer aided diagnosis, and for clinical routines or surgical planning. Furthermore, algorithms for segmentation of tubular structures are the key components of automated radiological diagnostic systems. For this reason it's important to obtain as much of the fine detail as possible. Therefore, automatic and robust methods are needed to extract the exact shape of the tubular structures. Moreover, when the vascular anatomical models have to be used in blood flow simulations, to provide clinicians with further insight into the medical analysis, it is important to construct embedded boundary representations of these models on the same computational grid used by the computational fluid dynamics simulations.

In this work, we mainly address the extraction of tubular structures from 2D/3D images, and we apply the proposed segmentation method for blood vessels, neurovascular

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 224–241, 2010.

structures and similar characteristics medical images. Ideally the segmentation model should be able to follow tubular structures and connect eventually disconnected parts.

Recently, several methods based on deformable models have been proposed for the segmentation of vessels or, more generally, tube-like structures. These model-based techniques can naturally capture the physics and geometry of shapes varying in space and time. We refer the reader to [13] for an extended review on vessel segmentation algorithms. Since explicit deformable model representation is usually impractical, level set techniques to evolve a deformable model have been recently introduced, which provide implicit representation of a deformable model. A curve in 2D or a surface in 3D evolves in such a way as to cover a complex shape or structure. Its initialization can be either manual or automatic and it needs not to be close to the desired solution. A disadvantage of level sets segmentation approach is the computational effort required to cover the entire domain of interest which is, in general, one dimension higher than the original one. Interested readers are referred to recent literature on the level set segmentation strategy for tubular structures [12], [6], [5], [21], [3], [10], [27], [1].

The method we propose can be considered as a geometric deformable model for the segmentation of tubular-like structures and it is characterized mainly by two components: the mean curvature flow and the directionality of the tubular structures. The major advantage of this technique is the ability to segment twisted, convoluted, and occluded structures without the user interactivity, following branching of different layers, from thinner to larger structures. One of the major disadvantages of the geometric deformable models, that is the computational cost, is strongly reduced by the proposed numerical approach limiting the number of unknowns in the solution of the linear systems involved in the evolution. The dependence on the grid resolution chosen to solve the discretized PDE model is still an open problem, and it should be chosen according to the minimum structure width to be segmented and eventually the resolution of the computational grid used by the computational fluid dynamics simulations.

An application of a variant of the proposed PDE model to the challenging problem of composed segmentation is given in [9], where the authors focused on an a priori composition of sources to obtain a single segmentation result according to a specific logic combination. Basically, the two PDE models presented in [9] and in the present work differ both for the initial function which is in the former computed as a Boolean composition of signed distance functions, and for the diffusion function which controls the evolution. In fact, while in [9] the anisotropic diffusion is defined only in the 2D case, this work extends the isotropic 3D PDE model presented in [9], providing an anisotropic model for 3D segmentation which is realized by an original finite volume discretization of the diffusion tensor.

The paper is organized as follows. The non-linear PDE model for tubular structures segmentation is introduced in Section 2, and an anisotropic variant is discussed in Section 3. Numerical aspects related to the discretization of the PDE model are discussed in Section 4. Its implementation by a suitable dynamic segmentation algorithm is discussed in Section 5. In Section 6 a speed up strategy is proposed to significantly reduce the computational complexity in the solution of the linear systems. Synthetic and real tests are provided in Section 7. Section 8 contains concluding remarks.

2 A Segmentation Model for Tubular Structures

In this section we formulate the segmentation problem as a geometric deformation of a 3D manifold driven by the properties of the structures we want to recover. Unlike parametric deformable models based on parametric surfaces deforming under the influence of internal and external forces, geometric deformable models use surfaces which propagate under a curvature dependent speed function, and are represented implicitly as higher dimensional functions. The main advantage of the implicit approach is the possibility to handle complex interfaces and adaptable changes in the topology during the development. However, these models can suffer some drawbacks, such as an over-smoothed recovery of the structures, uncomplete boundaries reconstruction, and ambiguous behaviors in case of long and thin objects where the propagation can flow over the boundaries.

To alleviate the problem of over-smoothed reconstruction, a well-known and successful strategy has been proposed by Perona and Malik, see [18], which detects structure boundaries in images by the diffusion function

$$g(s) = 1/(1+(s/\rho)^2), \tag{1}$$

where $\rho > 0$ is a small positive constant, and s represents the norm of the gradient of the image. The effect of this diffusion function when applied to PDE-based image denoising models, is to provide restored images of high quality and with sharp edges.

This section describes the implicit representation of the segmentation surface, represented as a 3D manifold which is the graph of a trivariate function ϕ mapping an open set $\Omega \subset \mathbb{R}^3$ into $\mathbb{R}$. We refer the reader to [17] for a excellent introduction to dynamic implicit surfaces.

The problem of determining the surface that best fits the object boundary represented in a 3D image I, can be posed as a volume minimization problem with objective function

$$V_g := \int_\Omega g(|\nabla I|)dV \qquad dV = \sqrt{1+|\nabla\phi|^2}dxdydz, \tag{2}$$

where the metric g is defined by (1) in Ω and V_g represents the weighted volume of a 3D manifold on Ω.

The local minimizer for the volume functional (2) which is attracted to the boundary of the object is determined by the steepest descent of (2), namely

$$\frac{\partial\phi}{\partial t} = \sqrt{1+|\nabla\phi|^2}\nabla.\left(g(|\nabla I|)\frac{\nabla\phi}{\sqrt{1+|\nabla\phi|^2}}\right), \tag{3}$$

or, equivalently, in advection-diffusion form

$$\frac{\partial\phi}{\partial t} = g(|\nabla I|)\nabla.\left(\frac{\nabla\phi}{\sqrt{1+|\nabla\phi|^2}}\right) + \nabla g\cdot\nabla\phi. \tag{4}$$

The PDE model (4) represents the mean curvature motion of the 3D manifold in 4D space with metric g. The metric g in (4) is the edge function appropriately chosen so

that the object boundaries act as attractors under a particular flow. This term allows us to extract sharp features, such as edges, corners, spikes, and to accelerate the deformation of the initial function. In the evolution of ϕ according (4) the 3D manifold assumes constant values for most regions far from the boundaries. The first term in (4) corresponds to a minimal volume regularization weighted by the function g, while the second term corresponds to the attraction to the image edges. The advection term in equation (4) introduces a driving force which moves the level surfaces towards the object boundaries.

In [16], a variant to the equation (4) is proposed for dealing with the boundary completion problem. The authors introduce in (3) a parameter $\varepsilon \in (0,1]$, with the effect to segment boundaries which are eventually uncompleted due to, for example, noise or corruptions in the acquisition phase. Then the PDE model (3) takes the form:

$$\frac{\partial \phi}{\partial t} = \sqrt{\varepsilon^2 + |\nabla \phi|^2} \nabla . \left(g(|\nabla I|) \frac{\nabla \phi}{\sqrt{\varepsilon^2 + |\nabla \phi|^2}} \right), \tag{5}$$

where the variability in the parameter ε provides both a regularization effect and a hole filling strategy. The PDE model (5) reduces to (3) for $\varepsilon = 1$. However, this variant does not improve the reconstruction of slightly disconnected tubular structures. The latter problem can be solved by the introduction of a suitable diffusion tensor, which is discussed in Section 3.

The model (5) is based on the use of Riemannian mean curvature flow of graphs to compute the segmentation results [22], unlike from other level set methods used in image segmentation (e.g geodesic active contour, see [2]) which follow the evolution of a particular level set of ϕ. In [16] the initial segmentation function is located in an approximate object center and then the function is driven by equation (5) and evolves to a numerical steady state. Its shock profile gives the segmentation result and shape of the object. To that goal, we choose a suitable isoline of the shock profile which represents the boundary of the segmented object. This isoline is most naturally taken as the average of maximal and minimal value of the final segmentation function.

Moreover, the ε-regularization was not used just as a tool to prove existence of a viscosity solution of the level set equation, but ε represents a modelling parameter. For small ε, the segmenting surface closes gaps in image object boundaries and is stabilized, i.e. almost unchanging by further evolution, so it is easy to stop the segmentation process.

The initialization of the evolving surface is usually a problem since it involves user interaction for locating in the 3D image some starting points at particular recognizable parts of the structure to be segmented. This is overcome by our method which automatically initialize the surface evolution using a suitable designed distance function, as described in Section 5.

3 An Anisotropic Variant of the Segmentation Algorithm

In this section we present a variant of the isotropic model (5) designed to improve significantly the connectivity of the coherent structures in the segmentation.

The idea is to evolve the segmentation function by isotropic mean curvature motion in homogeneous regions, and by anisotropic deformation driven by the directional field representing the structure orientation, in presence of tubular structures. We aim to capture the vessel's structure and the vessels directions locally by a local spatial coherence descriptor. Coherence enhancing image smoothing has been introduced by [25], [26] and successfully applied in image filtering by anisotropic diffusion. This type of nonlinear diffusion includes the construction of a diffusion tensor which can be directly built as follows.

Given an intensity function I, its Gaussian-smoothed version ∇I_σ, provides the contrast of an edge, but it is unsuited for finding closed parallel structures. Let us construct a matrix D_0 from the system of orthonormal eigenvectors,

$$\mathbf{v}_1 \| \nabla I_\sigma, \quad \mathbf{v}_2 \perp \nabla I_\sigma, \quad \mathbf{v}_3 \perp \nabla I_\sigma \quad \text{and} \quad \mathbf{v}_3 \perp \mathbf{v}_2 \tag{6}$$

which has the following form

$$D_0 = [\,\mathbf{v}_1\ \mathbf{v}_2\ (\mathbf{v}_1 \times \mathbf{v}_2)\,] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ (\mathbf{v}_1 \times \mathbf{v}_2)^T \end{bmatrix}. \tag{7}$$

The corresponding eigenvalues are computed so that diffusion along edges is preferred over diffusion across them, that is, by

$$\begin{aligned} \lambda_1 &= g(|\nabla I|) \\ \lambda_2 &= \begin{cases} g(|\nabla I|) & \text{if} \quad c = 0 \\ g(|\nabla I|) + (1 - g(|\nabla I|))e^{-K/c} & K > 0, \text{ else} \end{cases} \end{aligned} \tag{8}$$

where $g(\cdot)$ is defined in (1) and suitably adapts its values to the anisotropy, $c := |\nabla I|^2$, and K has the role of a threshold parameter. The diffusion tensor is a smooth, symmetric and positive definite matrix. The positive definiteness of the matrix D_0 follows easily from (8) and from the positivity of $g(\cdot)$.

The diffusion tensor is then obtained by applying a componentwise convolution with a Gaussian K_δ:

$$D := K_\delta * D_0 \tag{9}$$

where $\delta \geq 0$ represents the integration scale reflecting the characteristic size of the structures.

In local homogeneous areas of an image the diffusion is reduced to be the isotropic mean curvature motion driven by (5), in fact, following (8), D is replaced by $g(|\nabla I|)$. Areas nearby elongated structures are characterized by values of $g(\cdot)$ approaching to zero. The effect on the diffusion of the segmentation function is thus stronger along the coherence directions. In all the experiments reported in Section 7 we set $K = 1 \cdot 10^{-5}$. Since if $c >> K$ then $\lambda_2 \approx 1$, while if $c << K$ then $\lambda_2 \approx g(\|\nabla I\|)$, the segmentation function flows along the coherence direction when it approaches to the edges and stops when the object boundary is reached.

There are other possible approaches to construct and use the 3D structure tensor, see e.g. [25], [26]. We have described a simpler procedure according to the fact that

our tensor matrix is computed only at the pre-processing step and to our consequent assumption that the 3×3 matrices are constant at every image voxel.

If we consider the diffusion tensor D, computed by (9), in the segmentation model (5), we have the following non-linear anisotropic segmentation

$$\frac{\partial \phi}{\partial t} = \sqrt{\varepsilon + |\nabla \phi|^2} \nabla . \left(D \frac{\nabla \phi}{\sqrt{\varepsilon + |\nabla \phi|^2}} \right), \quad \phi(x,0) = \phi^0(x) \quad x \in \Omega, \tag{10}$$

where D does not depend on ϕ but only on I, therefore it is computed only at the pre-processing step. We will refer to the models (5) and (10) by isotropic and anisotropic segmentation models, respectively.

In the next section we illustrate a discretization of the PDE model (10) in terms of finite volume schemes.

4 Solving the PDE Model

The computational method for solving (10) is based on an efficient semi-implicit co-volume scheme suggested in [16], [4] suitable for image processing applications.

Equation (10) is accompanied with Dirichlet boundary conditions

$$\phi(x,t) = 0 \quad \text{in } [0,T] \times \partial\Omega, \tag{11}$$

and initial conditions

$$\phi(x,0) = \phi^0 \quad \text{in } \Omega, \tag{12}$$

where $[0,T]$ is a time interval, and $\Omega \subset \mathbb{R}^3$ is the computational domain which includes the tubular structures to be reconstructed. We assume that initial state of the function is bounded, i.e. $\phi^0 \in L_\infty(\Omega)$.

The semi-implicit in time discretization is obtained by treating the non-linear terms of the equation from the previous time step while the linear ones are considered on the current time level. Time-discretization of (10) by Euler's method yields the following semi-discrete scheme.

Let τ be a uniform discrete time step, ϕ^0 be a given initial function. Then, for every discrete time step $t_n = n\tau$, $n = 1, \ldots N$, we look for a function ϕ^n, solution of the equation

$$\frac{1}{\sqrt{\varepsilon^2 + |\nabla \phi^{n-1}|^2}} \frac{\phi^n - \phi^{n-1}}{\tau} = \nabla . \left(D \frac{\nabla \phi^n}{\sqrt{\varepsilon^2 + |\nabla \phi^{n-1}|^2}} \right). \tag{13}$$

The computational domain is obtained through Ω decomposition into cubic cells (corresponding to the image voxel structure) and the construction of a co-volume mesh using a complementary 3D tetrahedral grid. For each voxel boundary face we construct 4 tetrahedras using diagonals of the voxel boundary face. Two nodes of every tetrahedron correspond to centers of neighbouring voxels (the only degree of freedom nodes), and the other two nodes correspond to voxel boundary vertices. Then the co-volume (dual) mesh consists of cells p, $p = 1, .., M$, associated with the center of the voxels.

Let an initial function ϕ^0 be given by discrete values in the voxel centers, a piecewise linear approximation ϕ_h of the solution ϕ is built on the tetrahedral grid and it has constant value of gradients in tetrahedras.

By this construction the co-volume mesh corresponds exactly to the voxel structure of the image inside the computational domain Ω. Let C_p denote the set of all nodes q (centers of voxel) connected to the node p by an edge. This edge will be denoted by σ_{pq} and its length by h_{pq}. Then the co-volume boundary face e_{pq} is shared by the co-volumes p and q. We denote by $\mathscr{E}_{pq}$ the set of four tetrahedras having σ_{pq} as an edge. For each $T \in \mathscr{E}_{pq}$ let c_{pq}^T be the area of the portion of e_{pq} that is in T, i.e., $c_{pq}^T = m(e_{pq} \cap T)$, where m is measure in $\mathbb{R}^{d-1}$.

Let $\mathscr{N}_p$ be the set of the all tetrahedras that have p as a vertex. We will denote by $|\nabla\phi_T|$ the constant value of $|\nabla\phi_h|$ on $T \in \mathscr{T}_h$ and we will use the notation

$$|\nabla\phi_T|_\varepsilon = \sqrt{\varepsilon^2 + |\nabla\phi_T|^2}, \tag{14}$$

to define the regularized gradients.

Following the classical finite volume methodology [14] [8], we integrate (13) over every co-volume p, $p = 1,\ldots,M$ and using divergence theorem we get

$$\int_p \frac{1}{|\nabla\phi^{n-1}|_\varepsilon} \frac{\phi^n - \phi^{n-1}}{\tau} dx = \int_{\partial p} \frac{D_p \nabla\phi^n}{|\nabla\phi^{n-1}|_\varepsilon} \cdot \nu \tag{15}$$

where ν denotes the unit normal to ∂p, and D_p is the tensor matrix in p pre-computed by (9).

We refer the reader to [7] for the numerical aspects concerning the discretization of a similar 3D diffusion tensor by semi-implicit schemes in time and finite volume in space. The diffusion tensor introduced in [7] is used to construct a coherence enhancing image denoising filter.

We denote by $\omega_T = D_p \nabla\phi^n$ the vector obtained by transforming the gradient vector defined on tetrahedron $T \in \mathscr{T}_h$, using the matrix D_p associated with the cell p. The neighboring cells q of the co-volume p are in the set $C_p = \{q_w, q_e, q_s, q_n, q_t, q_b\}$ according to their position with respect to p as shown in Figure 1. According to the position of $q \in C_p$, the normal vector ν to the boundary of the co-volume p is $\nu = \nu_e = (1,0,0)$ for q_e, $\nu = \nu_w = (-1,0,0)$ for q_w, $\nu = \nu_t = (0,1,0)$ for q_t, $\nu = \nu_b = (0,-1,0)$ for q_b, $\nu = \nu_s = (0,0,1)$ for q_s and $\nu = \nu_n = (0,0,-1)$ for q_n.

For the approximation of the right hand side of (15) we get

$$\sum_{q \in C_p} \left(\sum_{T \in \mathscr{E}_{pq}} c_{pq}^T \frac{\omega_T}{|\nabla\phi_T^{n-1}|_\varepsilon} \right) \cdot \nu, \tag{16}$$

therefore, for each co-volume $q \in C_p$, the product $\omega_T \cdot \nu$ reduces to the scalar product

$$\begin{array}{l} \omega_T \cdot \nu = d_i \cdot \nabla\phi_p, \quad i = 1,2,3, \quad for \quad \nu_w, \nu_t, \nu_s \\ \omega_T \cdot \nu = -d_i \cdot \nabla\phi_p, \quad i = 1,2,3, \quad for \quad \nu_e, \nu_b, \nu_n \end{array} \tag{17}$$

where d_i represents the ith row of the tensor matrix $D_p = [D_{kj}]_{k,j=1,2,3}$ defined in (9). We discretize $\nabla\phi_p$ by forward differences for each $q \in C_p$. For example, for q_e, we have $\omega_T \cdot \nu_e = d_1 \cdot (\phi_{q_e} - \phi_p, \phi_{q_t} - \phi_p, \phi_{q_s} - \phi_p)$. Therefore, (16) can be rewritten as

$$\sum_{q \in C_p} a_{pq}(\phi_q^n - \phi_p^n). \tag{18}$$

For example, for a_{pq_e}, we have

$$a_{pq_e} = \frac{c_{pq}}{h_{pq}}(D_{11} \sum_{T \in \mathscr{E}_{pq_e}} \frac{1}{|\nabla\phi_T^{n-1}|_\varepsilon} + D_{21} \sum_{T \in \mathscr{E}_{pq_t}} \frac{1}{|\nabla\phi_T^{n-1}|_\varepsilon} + D_{21} \sum_{T \in \mathscr{E}_{pq_s}} \frac{1}{|\nabla\phi_T^{n-1}|_\varepsilon}). \tag{19}$$

Similarly we obtain the other coefficients $a_{pq}, \forall q \in C_p$.

The left hand side of (15) is approximated by

$$b_p^{n-1} \frac{\phi_p^n - \phi_p^{n-1}}{\tau} \tag{20}$$

where $b_p^{n-1} = M_p^\varepsilon m(p)$, $m(p)$ is measure in $\mathbb{R}^d$ of co-volume p and M_p is an approximation of the capacity function inside the finite volume p. We used the averaging of the gradients proposed by Walkington [24], i.e.

$$M_p^\varepsilon = \frac{1}{|\nabla\phi_p^{n-1}|_\varepsilon}, \quad |\nabla\phi_p^{n-1}|_\varepsilon = \sum_{T \in \mathscr{N}_p} \frac{m(T \cap p)}{m(p)} |\nabla\phi_T^{n-1}|_\varepsilon. \tag{21}$$

Applying boundary and initial conditions given by (11) and (12), respectively, and (18)-(20) at the time step $n, n = 1, \ldots, N$, we look for $\phi_p^n, p = 1, \ldots, M$, satisfying

$$b_p^{n-1}\ \phi_p^n + \tau \sum_{q \in C_p} a_{pq}^{n-1}(\phi_p^n - \phi_q^n) = b_p^{n-1}\ \phi_p^{n-1}. \tag{22}$$

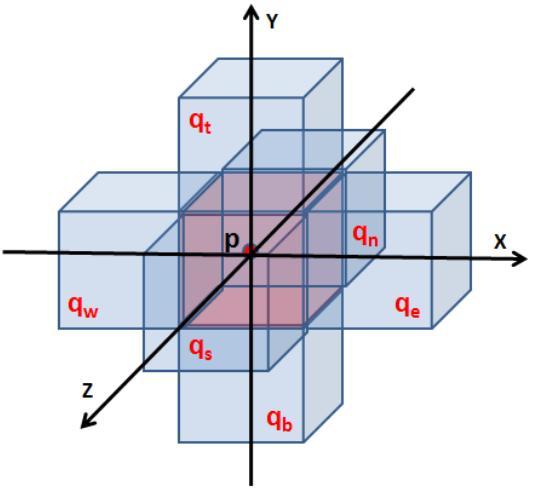


Fig. 1. Neighboring cells $q \in C_p$ of the co-volume p

That is a system of linear equations which can be written in matrix-vector form as

$$A\Phi^n = b, \tag{23}$$

where $A \in \mathbb{R}^{M\times M}$ is the coefficient matrix and $\Phi^n = (\phi_1, \ldots, \phi_M)$ is the vector solution.

We remark that the isotropic model (5), where D is replaced by $g()$, leads to a symmetric and diagonally dominant M-matrix A in (23), while the anisotropic model (10) cannot guarantee that the a_{pq} coefficients are all positives, and thus the matrix A is no longer an M-matrix. Any efficient iterative solver can be applied to solve the linear system (23) in case of sparse, symmetric, diagonally dominant M-matrices, for example we used the PCG (preconditioned conjugate gradient), [20], with an incomplete Cholesky factorization as effective tool to construct efficient preconditioners [11]. For the solution of the anisotropic model (10) since any theoretical consideration on the coefficient matrix A is still an open issue, in order to find out the solution even in case of singular matrix, we apply a variant of GMRES, which can overcome breakdowns of the Arnoldi process. GMRES is a popular iterative method for the solution of large linear systems of equations with a square nonsingular matrix [20]. The standard implementation is based on the application of k steps of the Arnoldi process to the matrix A with initial vector b. When the matrix is singular, GMRES may break down before an acceptable approximate solution has been determined. In [19] the authors described an efficient technique to overcome breakdown and hence to work out an approximation of the solution also in the singular case, they refer to this strategy as breakdown free GMRES (*BFGMRES*). We observe that in all the experiments run we have never found singularities in the matrices A.

A speed up strategy to solve the linear systems (23) is discussed in Section 6.

5 The Dynamic Segmentation Algorithm

In this section, we discuss the implementation of the PDE model (10) for the segmentation of 3D images. The same algorithm can easily be adapted to the segmentation of 2D images.

A gray-level 3D image I containing tubular structures is thresholded to obtain a rough segmentation estimate which is used to generate an initial distance function. This represents the initial surface which continuously evolves, by following equation (10), towards the boundaries of the tubular structures. The evolution is stopped when a change of solution in time (in L_2-norm) is less than a prescribed tolerance.

The thresholding process consists of making a binary image I_0 from the input gray-scale image I based on a user threshold. Since the images can be represented by a real-valued function defined on a region $\Omega \subset \mathbb{R}^3$ or by the discretization of such a function, the threshold process allows us define in a natural way the sub-domain $\Omega_1 \subset \Omega$, $\Omega_1 := \{x \in \Omega : I_0(x) = 1\}$. The choice of the threshold value is not an issue because of the fact that in practical cases the structures to be segmented are characterized by a particular intensity value represented by a given gray level. For example, vascular system containing a contrast fluid, can be identified as the brightest formations in the volume. As it will be better illustrated by the examples in Section 7 the 3D mask I_0 resulting from the pre-computed binarization, in general, preserves the largest tubular

structures while breaks up into small pieces the thinnest ones. These structures will be perfectly recover and reconnected by the surface evolution process.

Given I_0, we define the initial function as

$$\phi^0(x) = \begin{cases} 1 & \text{if} \quad x \in \Omega_1 \\ (1 - \frac{dist(x,I_0)}{max\{dist(x,I_0)\}}) & \text{else} \end{cases}, \tag{24}$$

where $dist(\cdot)$ represents the Euclidean distance implemented by $distance\_function(I_0)$ in Tubular Segmentation Algorithm.

The dynamic segmentation method is described by the following Tubular Segmentation Algorithm which applies equation (10) to a given input image I, and uses the input parameter $Thrs$ as threshold value for the initial binarization.

Tubular Segmentation Algorithm

INPUT: I, $Thrs$, Tol
OUTPUT: segmented surface Γ
STEP **pre-comp**:
    $I_0 = threshold(I, Thrs)$
    $\phi(x,0) = distance\_function(I_0)$;
    Compute the tensor matrix D;
STEP **evolve**:
    repeat
    compute $\phi(x,t_n)$ *in* (10) *by solving* (23) *for* Φ, $\forall x \in \Omega_1$;
    until $\|\phi(x,t_n) - \phi(x,t_{n-1})\|_2 < Tol$;
STEP **post-comp** Γ *is extracted as the s-level set of* ϕ

In the final STEP `post-comp`, the reconstructed surface is obtained from the implicit surface ϕ as the zero level set of the function $\phi(x) - s$, that is the s-level set of ϕ: $\{x \in \Omega : \phi(x) = s\}$, where $s = (max(\Phi) + min(\Phi))/2$. This is motivated by the fact that the flow driven by (10) forms a sharp step in the proximity of the object boundaries, while it approaches at constant values inside/outside the object.

6 Fast Linear System Resolution

The computation of an approximate solution of the PDE non-linear model (10) requires the update of the unknown function $\phi(x,t)$ for each co-volume of the 3D image domain Ω, at each time step n, which is a time consuming process. Since the unknown function $\phi(x,t)$ evolves only on nodes sufficiently close to the structure boundary, we speed up the computation by determining the updated values for $\phi(x,t)$ only for the nodes x identified by $\phi^0(x) > \eta$, for a given small tolerance η. The idea of computational adaptivity goes back to the narrow band level set methods in [23], but the way this can be realized strongly depends on the model discretization and on the iterative linear system considered.

In case of vessel structures, for example, this means a significant reduction of the computational effort required since the number of nodes representing the vessels is small compared with the dimension of the entire 3D image which contains them.

In practice, at each time step, the number of unknowns of the linear system (23) is significatively reduced, and thus both the storage and the computational cost are much lower. Since at each row of A corresponds a node in Ω, considering a limited number of nodes $M_1 << M$, we get a linear system with a sparse coefficient matrix which contains $M - M_1$ zero rows. Moreover, if we consider to apply a Krylov-type iterative method, like PCG or GMRES methods, where the main computational step involves a matrix-vector product with a vector of unknowns which contains $M - M_1$ zeros, then we end up with a matrix A which has $M - M_1$ zero rows and columns.

It is easy to verify that applying a suitable permutation to rows and columns of A, and corresponding elements of b, we get a linear system with a coefficient matrix $\widetilde{A} \in \mathbb{R}^{M_1 \times M_1}$ with the same epta-diagonal structure as A. In a similar way, the same permutation applied to the components of the right-hand side vector b, leads to a vector $\widetilde{b} \in \mathbb{R}^{M_1}$.

Therefore, instead of solving the linear system (23) which involves a $M \times M$ coefficient matrix, we apply a Krylov-like iterative method for computing the solution $\widetilde{\Phi}$ of the linear system

$$\widetilde{A}\Phi = \widetilde{b}. \tag{25}$$

Thus the coefficient matrix $\widetilde{A}$ is obtained by considering only the nodes close to the boundary of the tubular structures. Moreover, when it has been computed as the discretization of the isotropic model, $\widetilde{A}$ has full rank and it is symmetric positive definite, thus the PCG method is used instead of the bfgmres algorithm.

In general, iterative methods for solving linear systems (23) and (25) require to store only the non-vanishing elements of the coefficient matrix and to compute at each iterative step a matrix-vector multiplication. A very simple test on the distance function $dist()$ let us determine the elements of the matrix $\widetilde{A}$ and vector $\widetilde{b}$, moreover the multiplication is inexpensive because of the lower dimension matrix. This strategy considers only a small portion Ω_1 of the grid domain Ω with a negligible penalization on the results.

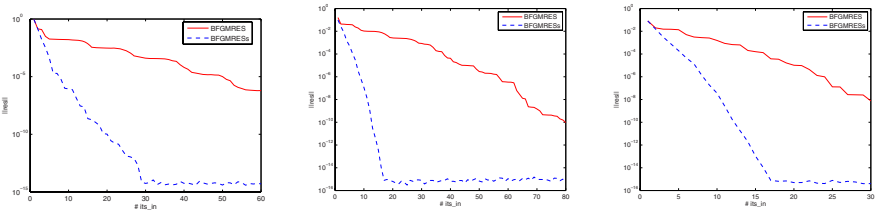
In order to estimate the error between the approximated solutions $\widetilde{\Phi}$ and Φ we compare the results obtained by solving the linear systems (23) and (25) using the bfgmres algorithm with the stopping criterium driven by the tolerance $1 \cdot 10^{-6}$, for three 3D simple test images.

Each 3D image consists of 10^3 voxels, representing a black cube in a white background, with cubes with increasing dimensions. The dimension of the cube determines the number of unknowns in (25). In Table 1 and in Fig.2 we illustrate the results for the three 3D test images. The results displayed in Table 1 columns 2-3 (Fig.2 (a)) are related to 3D image with cube dimension 4, in Table 1 columns 4-5 (Fig.2 (b)) the cube dimension is 6, while in Table 1 columns $6-7$ (Fig.2 (c)) the cube-dimension is 7. All the coefficient matrices have full rank.

In Table 1, for each of the three examples, the column denoted by *BFGMRES* represents the solution of the linear system (23), while the column labelled *BFGMRESs* reports the results obtained by solving (25). The rows in Table 1 display the dimensions

Table 1. Resolution of the linear systems (23) and (25) by BFGMRES iterative solver. The Table reports the relative errors ($\|err\|$), the residual norms ($\|res\|$), the number of iterations ($\#its_{in}$), and the number of non-vanishing elements (nz) for matrices of dimension dim and full rank.

	BFGMRES	BFGMRESs	BFGMRES	BFGMRESs	BFGMRES	BFGMRESs
dim	1000	144	1000	343	1000	512
nz	6400	1008	6400	2401	6400	3584
$\#its_{in}$	55	9	74	9	24	9
$\|res\|$	$9.2 \cdot 10^{-7}$	$9.8 \cdot 10^{-7}$	$6.3 \cdot 10^{-7}$	$6.2 \cdot 10^{-7}$	$9.8 \cdot 10^{-7}$	$1.8 \cdot 10^{-7}$
$\|err\|$	-	$2.1 \cdot 10^{-7}$	-	$1.8 \cdot 10^{-7}$	-	$6.9 \cdot 10^{-7}$



(a) (b) (c)

Fig. 2. Residual errors as function of the number of iterations required by BFGMRES iterative method used to solve linear systems (23) (solid curve) and (25) (dashed curve) for the three 3D test images; (a) case 3D image with cube dimension 4, (b) case 3D image with cube dimension 6, (c) case 3D image with cube dimension 7

of the matrices $A \in \mathbb{R}^{M \times M}$ and $\widetilde{A} \in \mathbb{R}^{M_1 \times M_1}$ denoted by dim, the non-vanishing elements by nz, the norm of the residuals, $\|res\|$, the relative errors on the approximate solutions

$$\|err\| = \|\boldsymbol{\Phi} - \widetilde{\boldsymbol{\Phi}}\|_2 / \|\boldsymbol{\Phi}\|_2,$$

and the number of iterations ($\#its_{in}$) required to solve the linear systems by the bfgmres method. The results are qualitatively the same from a visual inspection, and the relative errors reported in Table 1 confirm the goodness of the approximate solutions.

While in Table 1 we report the results related to a single time step, in Table 2, we consider the relative errors in the approximate solution $\widetilde{\boldsymbol{\Phi}}$ for an increasing number of time iterations ($\#its_{out}$) in STEP `evolve` of Tubular Segmentation Algorithm. At each

Table 2. Relative errors in the approximate solution ϕ as functions of the number of outer iterations ($\#its_{out}$)

$\#its_{out}$	$\|err\|$
2	$2.1 \cdot 10^{-5}$
4	$2.4 \cdot 10^{-5}$
6	$2.5 \cdot 10^{-5}$
8	$2.7 \cdot 10^{-5}$
10	$2.8 \cdot 10^{-5}$

time step the dimension of the linear systems (25) remains the same, and the error on the solution propagated by the function evolution is limited in time.

The computed examples in the next section demonstrate the effectiveness of the proposed anisotropic segmentation method. The examples deal with some of the challenges in vasculature segmentation techniques, which include the ability to track the vessels to the second and third layers of branching and the possibility to grow and reconnect gaps while keeping separated thinner long tubular structures which are very close to each other.

7 Experiments and Results

In this section we present 2D and 3D examples which demonstrate the performance of the proposed segmentation algorithm in case of synthetic and real images. The synthetic 2D examples have been constructed with the aim to point out the influence of the diffusion tensor introduced in model (10) in the segmentation of tubular structures.

We applied Tubular Segmentation Algorithm, with or without diffusion tensor, to each of the example images and we set the parameters involved to $\tau = 1 \cdot 10^{-3}$, time step for 3D images and $\tau = 1 \cdot 10^{-2}$ time step for the 2D images, and $\varepsilon = 1 \cdot 10^{-3}$.

In the first 2D example we are concerned with the segmentation of a carotid vascular system represented in a Magnetic Resonance Angiography (MRA) image made up of 182×182 pixels shown in Fig. 3 (left). MRA technology is based on detection of signals from flowing blood and suppression of signals from other, static, tissues. The blood vessels appear as high intensity regions in the image. The structures to be segmented represent several vessels of variable diameters which are close to each other, partial occlusions and intersections make the segmentation very challenging. In Fig. 3 the segmented structures obtained by 10 iterations of the isotropic (center) and anisotropic

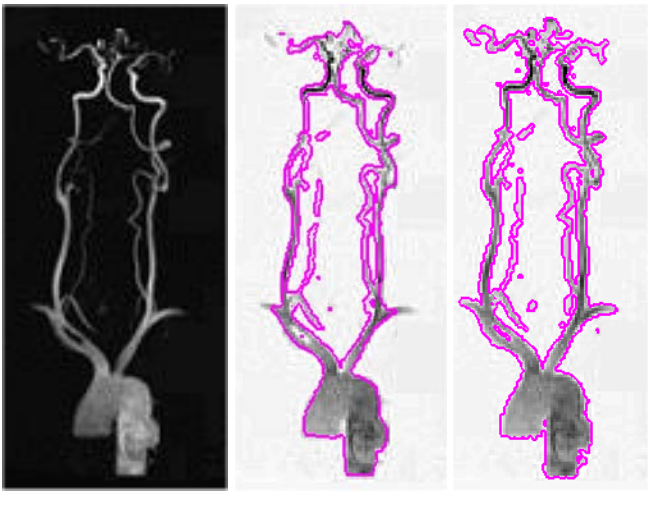


Fig. 3. Carotid vascular system segmentation. The original image (left); segmentation by the isotropic segmentation model (center), segmentation by the anisotropic model (right).

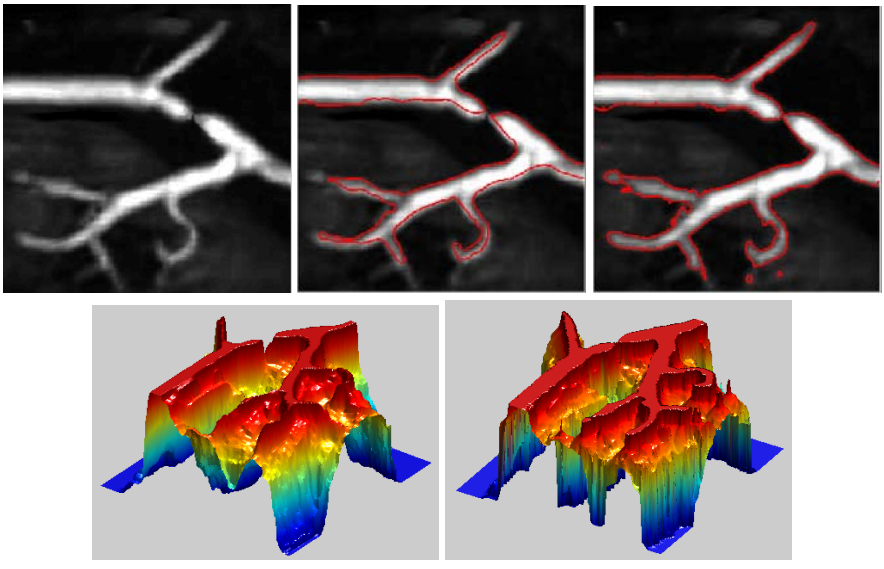


Fig. 4. Kidney vascular system segmentation. First row: original image to be segmented (left); results obtained by the isotropic segmentation model (center), and anisotropic segmentation model (right). Second row: the segmentation function ϕ is shown at the final time step for isotropic (left) and anisotropic (right) evolutions.

(right) segmentation models are displayed. The segmented curves in the second row are extracted using contour values $s = 0.90$ and $s = 0.98$, respectively. Visual comparison shows the anisotropic segmentation method to give the most accurate restoration.

The second example illustrates the ability to reconstruct structures which present small occlusions along the coherence direction. We would like to determine the boundary curves of the kidney vascular system represented in the MR image shown in Figure 4 (first row). The image consists of 220×220 pixels. Applying 10 time steps of the anisotropic segmentation model, the structure is well reconstructed while maintaining the narrowing, as shown in Fig. 4 (second row, right). The propagation driven by the isotropic segmentation model leads to enhance the disconnections, as shown in Fig. 4 (second row, left). The iso-contours determined by the segmentation functions shown in Fig. 4 (second row), using $s = 0.95$ are over-imposed on the original image and shown in Fig. 4 (first row).

In order to investigate the capability to segment two distinct tubular structures which are very close along the flow direction, we built up the synthetic image of dimension 200×200 pixels shown in Fig. 5 (first row, left). The segmentation result after 10 iteration steps of the isotropic segmentation model are illustrated in Fig. 5 (first row, center), while the result obtained by the anisotropic segmentation model are displayed in Fig. 5 (first row, right). The two structures which are originally separated have been joined together by the isotropic approach while a very good reconstruction of the two tubular structures is obtained by the anisotropic model. The representation of the segmentation

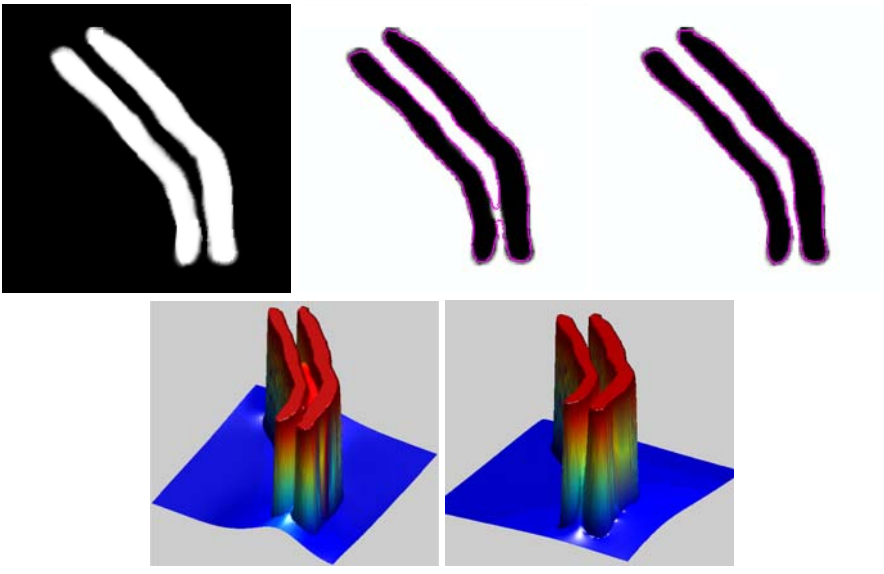


Fig. 5. Segmentation of structures in synthetic image. First row: the original image to be segmented (left); results obtained by the isotropic segmentation model (center), and anisotropic segmentation model (right). Second row: the segmentation function ϕ is shown at the final time step.

functions shown in the second row in Fig.5 let us to observe how the evolution of the isotropic function tends to fill the gap between the two structures.

We finally illustrate some preliminary experimental results on 3D real data sets from medical imaging.

In the first 3D example, we apply Tubular Segmentation Algorithm to a MRA image of pelvic vasculature shown by volume rendering in Fig.6(left). The `pelvic` data set considered consists of $181 \times 181 \times 181$ voxels extracted from a volume data set of $181 \times 383 \times 383$ voxels. In particular, the red box outlined in the image includes the region of interest inside the `pelvic` image considered for segmentation. The data sets `pelvic` and `brain`, used in the 3D examples, are DICOM sample images available on the web site (`http://www.osirix-viewer.com/Downloads.html`). The surface boundary obtained after 15 iterations of Tubular Segmentation Algorithm is displayed in Fig.6(right). Visual inspection shows a very good reconstruction after only a few time steps. We observe that the reconstruction of the thin vessels in the image is accurate even if these structures were not well defined by the initial 3D mask I_0. Computational time was 80sec for each iteration on a standard PC with 2GHz, and 4Gb RAM.

In the second example, volumetric segmentation is applied to MRA data set representing a brain aneurysms (vessel wall dilatations). The volumetric data set `brain` of dimension $120 \times 250 \times 200$ has been extracted from a $120 \times 448 \times 540$ MRA image of the brain-neck vasculature system shown in Fig.7(left), and presents vessel patterns with different curvatures, diameters and bifurcations. The final segmentation obtained

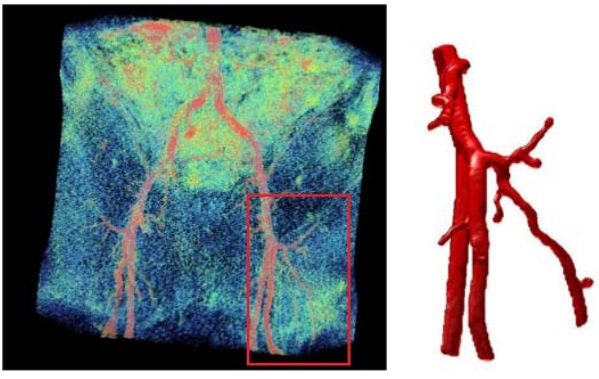


Fig. 6. Segmentation of the `pelvic` data set. The original MRA image (left); segmentation result (right).

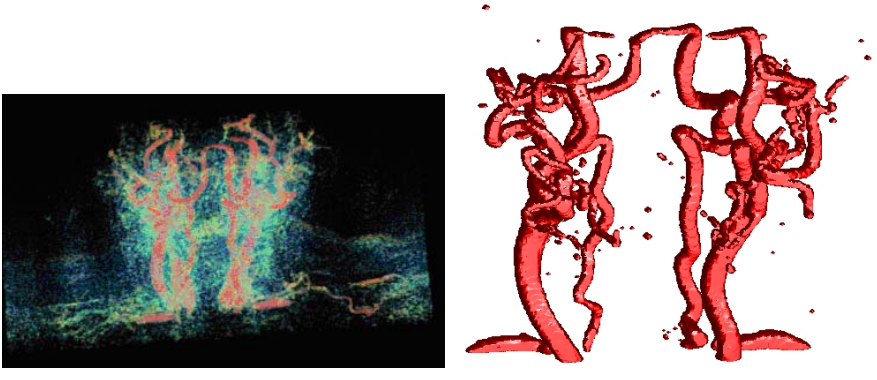


Fig. 7. Segmentation of the `brain` volume data set. The original MRA image (left); segmentation result (right).

after 10 time steps of Tubular Segmentation Algorithm is shown in Fig. 7 (right). The technique can also be used to compute vessel radii, a clinically useful measurements in case of aneurysms.

These preliminary results demonstrate the applicability of the developed method for the segmentation of quite different and topological complex tubular structures.

8 Conclusions and Future Work

In this paper we introduced a segmentation method based on the distance function which is successfully applied to 2D/3D images containing tubular structures. The algorithm is automatic, accurate and fast. The latter is due to a speed up strategy in the iterative method for linear systems. The algorithm is provided with a diffusion tensor to move the evolving surface toward elongated tubular structures, connecting gaps in

the underlying raw data, while keeping the structures distinguished along the coherence direction. Future work will concentrate mainly on theoretical results and the validation of the method, on the comparison of the results we obtained using the proposed algorithm with other segmentation techniques, with the support of experts in medical image analysis.

References

1. Antiga, L., Ene-Iordache, B., Remuzzi, A.: Computational geometry for patient-specific reconstruction and meshing of blood vessels from MR and CT angiography. IEEE Transaction on Medical Imaging 22(5), 674–684 (2003)
2. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Proc. ICCV 1995, Cambridge, MA, pp. 694–699 (1995)
3. Cohen, L.D., Deschamps, T.: Segmentation of 3D tubular objects with adaptive front propagation and minimal tree extraction for 3D medical imaging. Computer Methods in Biomechanics and Biomedical Engineering 10(4), 289–305 (2007)
4. Corsaro, S., Mikula, K., Sarti, A., Sgallari, F.: Semi-implicit co-volume method in 3D image segmentation. SIAM Journal on Scientific Computing 28(6), 2248–2265 (2006)
5. Dekanic, K., Loncari, S.: 3D vascular tree segmentation using level set deformable model. In: Proceeding of the 5th International Symposium on image and signal Processing and analysis (2007)
6. Deschamps, T., Schwartz, P., Trebotich, D., Colella, P., Saloner, D., Malladi, R.: Vessel segmentation and blood flow simulation using level set and embedded boundary method. International Congress Series, vol. 1268, pp. 75–80 (2004)
7. Drblikova, O., Mikula, K.: Semi-implicit Diamond-cell Finite volume Scheme for 3D Nonlinear Tensor Diffusion in Coherence Enhancing Image Filtering. In: Eymard, R., Herard, J.M. (eds.) Finite Volumes for Complex Applications V: Problems and Perspectives, ISTE and WILEY, London, pp. 343–350 (2008)
8. Eymard, R., Gallouet, T., Herbin, R.: The finite volume method. In: Ciarlet, P., Lions, J.L. (eds.) Handbook for Numerical Analysis, vol. 7. Elsevier, Amsterdam (2000)
9. Franchini, E., Morigi, S., Sgallari, F.: Composed Segmentation of Tubular Structures by an Anisotropic PDE Model. In: Tai, X.-C., et al. (eds.) SSVM 2009. LNCS, vol. 5567, pp. 75–86. Springer, Heidelberg (2009)
10. Gooya, A., Liao, H., et al.: A variational method for geometric regularization of vascular segmentation in medical images. IEEE Transaction on Image Processing 17(8), 1295–1312 (2008)
11. Handlovi cova, A., Mikula, K., Sgallari, F.: Semi-implicit complementary volume scheme for solving level set like equations in image processing and curve evolution. Numer. Math. 93, 675–695 (2003)
12. Hassan, H., Farag, A.A.: Cerebrovascular segmentation for MRA data using levels set. International Congress Series, vol. 1256, pp. 246–252 (2003)
13. Kirbas, C., Quek, F.: A review of vessel extraction techniques and algorithms. ACM Computing Surveys 36(2), 81–121 (2004)
14. Le Veque, R.: Finite volume methods for hyperbolic problems. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge (2002)
15. Meijering, E., Niessen, W., Weickert, J., Viergever, M.: Diffusion-enhanced visualization and quantification of vascular anomalies in three-dimensional rotational angiography: Results of an in-vitro evaluation. Medical Image Analysis 6(3), 215–233 (2002)

16. Mikula, K., Sarti, A., Sgallari, F.: Co-volume level set method in subjective surface based medical image segmentation. In: Suri, J., et al. (eds.) Handbook of Medical Image Analysis: Segmentation and Registration Models, pp. 583–626. Springer, New York (2005)
17. Osher, S., Fedwik, R.: Level Set Methods and Dynamic Implicit Surfaces. Applied Mathematical Sciences, vol. 53. Springer, New York (2002)
18. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. 12, 629–639 (1990)
19. Reichel, L., Ye, Q.: Breakdown-free GMRES for singular systems. SIAM J. Matrix Anal. Appl. 26, 1001–1021 (2005)
20. Saad, Y.: Iterative methods for sparse linear systems. PWS Publ. Comp. (1996)
21. Scherl, H., et al.: Semi automatic level set segmentation and stenosis quatification of internal carotid artery in 3D CTA data sets. Medical Image Analysis 11, 21–34 (2007)
22. Sarti, A., Malladi, R., Sethian, J.A.: Subjective Surfaces: A Geometric Model for Boundary Completion. International Journal of Computer Vision 46(3), 201–221 (2002)
23. Sethian, J.A.: Level Set Methods and Fast Marching Methods. Evolving Interfaces in Computational Geometry. Fluid Mechanics, Computer Vision, and Material Science. Cambridge University Press, Cambridge (1999)
24. Walkington, N.J.: Algorithms for computing motion by mean curvature. SIAM J. Numer. Anal. 33(6), 2215–2238 (1996)
25. Weickert, J.: Coherence-enhancing diffusion of colour images. Image and Vision Computing 17, 201–212 (1999)
26. Weickert, J., Scharr, H.: A scheme for coherence enhancing diffusion filtering with optimized rotation invariance. Journal of Visual Communication and Image Representation 13(1/2), 103–118 (2002)
27. Westin, C.-F., Lorigo, L.M., Faugeras, O.D., Grimson, W.E.L., Dawson, S., Norbash, A., Kikinis, R.: Segmentation by Adaptive Geodesic Active Contours. In: Proceedings of MICCAI 2000, Third International Conference on Medical Image Computing and Computer-Assisted Intervention, October 11-14, pp. 266–275 (2000)

Smoothing the Antagonism between Extraordinary Vertex and Ordinary Neighbourhood on Subdivision Surfaces

Cédric Gérot, François Destelle, and Annick Montanvert

GIPSA-lab, Grenoble, France

Abstract. Recent methods for tuning a subdivision scheme create a concentric wave pattern around the extraordinary vertex (EV). We explain it as resulting from the antagonism between the rules which would create a nice limit surface at the EV and the ordinary rules used in the surrounding regular surface.

We show that even a scheme which fulfils the most recently proposed conditions for good convergence at the EV may still produce this wave pattern.

Then, in order to smooth this antagonism, we define any new vertex as a convex combination of the ideal new vertex from the EV point of view and the one defined with ordinary rules. The weight of the extraordinary rules decreases as the new vertex is topologically farther from the EV.

The concentric wave pattern shades off whereas the expected conditions are not too much spoiled. This tuning method remains simple and useful, involving no optimisation process.

1 Introduction

Subdivision schemes are efficient tools for generating free-form surfaces and are used in many fields of computer graphics and geometric design [26]. Starting from a polygonal mesh of arbitrary connectivity, they define surfaces as the limit of a sequence of polygonal meshes each of whose vertices is a linear combination of a small number of vertices belonging to the previous mesh in the sequence.

Many schemes have been proposed for various lattices [8] and for interpolating or approximating the first mesh of the sequence. They all propose different sets of rules for defining a vertex of a subdivided mesh. One set is dedicated to the regular regions of a mesh. Another set defines the vicinity of extraordinary elements (vertices or faces). An extraordinary vertex (EV) does not have the ordinary number of incident edges, e.g. six for a triangular lattice. An extraordinary face has a non ordinary number of sides. Notice that usually no extraordinary vertex or face is added through subdivisions after the first step, isolating such extraordinary elements. Sometimes more sets are proposed for the vicinity of surface features such as crest lines or border.

It is easy to define rules generating convenient limit surfaces in regular regions. For instance, if an approximating surface is expected, the successively subdivided regular meshes are interpreted as control polyhedra of the same Box-spline surface, the definition of new vertices being derived from the associated uniform knot-insertion operator. The limit surface is this same Box-spline surface with an n-sided hole around each extraordinary element. Every hole is filled in using a patch derived from the extraordinary

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 242–260, 2010.

rules. More precisely, at each subdivision step, the extraordinary rules define the control polyhedron of a new Box-spline ring which continuously extends the regular part of the surface within the n-sided hole. Unfortunately, even if these n-sided patches connect continuously with the regular Box-spline surface, they may have an unsatisfactory shape such as pinching, a flat spot, or fluctuating curvature [21].

To solve these problems, two methods have been proposed. On the one hand, specific surfaces are used for these n-sided patches. Karčiauskas and Peters propose nested guided surface rings [10]. Levin proposes blending the limit surface with a low degree polynomial [12]. On the other hand, the method known as subdivision scheme tuning involves staying within the simple subdivision framework and modifying extraordinary rules to satisfy expected properties. The surfaces produced by solutions of the first kind may have a better shape than the ones produced by solutions of the second kind, however one of the main properties of subdivision schemes is lost: the simplicity of their implementation. As a consequence, we believe that it is worthwhile to continue investigating the second method and to propose rules to obtain the best limit surfaces within the simple framework of stationary subdivision schemes (the rules are the same whatever the subdivision step).

1.1 Related Work

The expected properties which the extraordinary rules should satisfy must lead to n-sided patches behaving as well as possible. Some studies on n-sided patches generated by subdivisions focus on their mathematical properties at the limit [18,16,25]. Achieving C^2-continuity is impossible for low-support stencils [19]. Other studies investigate their shape properties at the limit [5,15] or the convergence behaviour of the scheme [6] and provide conditions which can be used in practice for tuning subdivision schemes. In Sect. 2, the most recently proposed conditions for the good behaviour of generated surfaces are recalled.

Because the rules are linear combinations of vertices, they can be collected into one matrix called the *subdivision matrix*. Each line of this matrix, known as the *stencil*, collects the coefficients applied on old vertices to define a new one. Each column, known as the *mask*, collects the influence that an old vertex has on the definition of new ones. If the scheme is stationary, then the behaviour at the limit is characterised by the eigenstructure of the subdivision matrix.

The most common tunings define a specific stencil for the new EV only and in order to come close to few expected conditions of eigenvalues [4,13,9,23]. These schemes may produce artifacts localised at the EV like the pit in Fig. 3 (middle column). Only a few expected conditions can be fulfilled with such schemes because the only stencil for the new EV does not contain enough degrees of freedom. Other studies propose new stencils for more vertices than the EV (most commonly they define new stencils for the EV and the first ring of its neighbours). For tuning Loop's scheme, Prautzsch and Umlauf propose building the subdivision matrix from eigenvalues chosen in order to generate G^2 limit surface with zero curvature at the EV, and keeping the original eigenvectors [17]. But the zero curvature at the EV introduces a flat spot which is undesirable. Recently Ginkel and Umlauf proposed an extension to this study [7]. But they

leave the simple framework, proposing a non-stationary scheme which takes into account the geometry of the mesh. Loop presents a modified version of his scheme with bounded curvature and convex hull property [14]. The convex hull property is a sufficient but not a necessary condition for designing a limit surface which lies near the input mesh. Moreover, this scheme causes a concentric wave pattern around the EV. Barthe and Kobbelt use the most recent conditions on eigenvalues and eigenvectors [3]. But the coefficients of the stencils are computed as minimising a linear combination of a number of quality measures, and their scheme may also cause the same concentric wave pattern (see Fig. 10).

Peters and Reif highlighted the necessity to take into account not only the convergence at the EV, but also the shape of its neighbourhood [15]. They proposed with Karčiauskas using shape-at-the-limit charts for characterising the quality of a scheme [11]. Most recently, Augsdörfer et al. [1] extended their work by actually tuning schemes with similar charts. They proposed changing the mask of the EV. Some coefficients in the mask are chosen to satisfy conditions of eigenvalues. The others result from an optimisation process, minimising variation in curvature computed on a large set of input meshes.

1.2 Our Proposal

Like these last studies, we stress the fact that every tuning method should take into account not only conditions of eigenvalues and eigenvectors for a good convergence at the EV, but also the shape of the regular neighbourhood. Indeed, in Sect. 3 it is shown that even a scheme which fulfills all the most recently proposed conditions on eigenvalues and eigenvectors can still produce the same concentric wave pattern as those mentioned above. The subdivision matrix of such a scheme is built from the expected eigenstructure using the technique proposed by Prautzsch and Umlauf. We propose to explain this concentric wave pattern as resulting from antagonism between the behaviour of the shape expected for the EV and the one expected for the surrounding area. In order to achieve a compromise between these two behaviours, we blend in Sect. 4 the extraordinary rules with the ordinary ones. The weight of the extraordinary rules decreases as the new vertex is topologically farther from the extraordinary element. Finally, the coefficients of the resulting stencils are given as explicit equations depending on the valence of the extraordinary element and not as a result of an optimisation process, which is still difficult to set. As a result, the concentric wave pattern shades off whereas the expected conditions are not spoiled too much (see Sect. 6).

From now on, the article deals with the tuning of Loop's scheme: the lattice is triangular, the extraordinary element is a vertex whose valence is not equal to six, and the ordinary rules generate a quartic Box-Spline [13]. Its practical generalisation to other stationary schemes is discussed in Sect. 8.

2 Expected Properties

The properties one can expect from a stationary subdivision scheme are of two kinds. Firstly, the scheme must be simple and unambiguous to apply in practice, which leads to

symmetry conditions. Secondly, the generated limit surface must behave as well as possible, in particular around the EV, which leads to conditions of the subdivision matrix eigenstructure.

2.1 Symmetries

Before presenting the symmetry conditions of the stencil coefficients, let us introduce some notation.

Notation. We assume that the vicinity of the EV is made up of ordinary vertices. As a consequence, it can be divided into n topologically equivalent sectors, where n is the EV valence. Let us identify each sector with an integer $k \in \mathscr{K} := \{0,\dots,n-1\}$. Moreover the vicinity can be divided into rings of vertices topologically equidistant from the EV. Let us identify each ring with a letter $X \in \{A\} \cup \mathscr{X}$ with $\mathscr{X} := \{B,C,D\}$. For Loop's scheme, each vertex in the vicinity of interest is uniquely determined by a couple (ring, sector). Notice that the EV being alone in its ring, no sector number is necessary: A is sufficient. Let X_k be an old vertex and let x_k be the new one as illustrated in Fig. 1 for Loop's scheme.

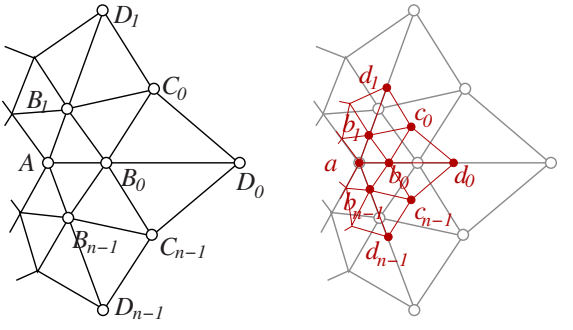


Fig. 1. Vertex notation around an EV

Stencils. If no feature, such as crest lines or borders must be taken into account, then one expects for the stencil to be independent from the choice of sector 0. This is called *rotational symmetry*. Moreover Loop's scheme does not rotate the lattice between two successive meshes in the subdivision sequence, the weight of X_{j+k} should be the same as the weight of X_{j-k} in the definition of y_j. This is called *mirror symmetry*.

Under these assumptions, the weight of X_{j+k} in the definition of y_j depends only on the rings X and Y and on the gap $\pm k$ between the sectors j and $j+k$ or j and $j-k$, which leads to the following stencil:

$$y_j = s_{y,A}^{n-j} A + \sum_{X \in \mathscr{X}, k \in \mathscr{K}} s_{y,X}^{k} X_{j+k} \tag{1}$$

with indices taken with modulo n and where $s^k_{y,X} = s^{n-k}_{y,X}$ are real scalars. Discrete Fourier techniques, used already by Doo and Sabin [5], allow us to write, if n is odd,

$$s^k_{y,X} = \frac{1}{n} S^0_{y,X} + \frac{2}{n} \sum_{\omega=1}^{(n-1)/2} S^\omega_{y,X} c^{\omega,k}_{\phi_X - \phi_Y} \tag{2}$$

and if n is even,

$$s^k_{y,X} = \frac{1}{n} S^0_{y,X} + \frac{2}{n} \sum_{\omega=1}^{(n-2)/2} S^\omega_{y,X} c^{\omega,k}_{\phi_X - \phi_Y} + \frac{1}{n} S^{n/2}_{y,X} e^k_{\phi_X - \phi_Y} \tag{3}$$

where

$$c^{\omega,k}_{\phi_X - \phi_Y} = \cos\left(\frac{2\pi\omega}{n}(k + \phi_X - \phi_Y)\right)$$
$$e^k_{\phi_X - \phi_Y} = \exp\left(\mathrm{i}\pi(k + \phi_X - \phi_Y)\right)$$

and if $\omega \neq 0$, $S^\omega_{a,X} = S^\omega_{y,A} = 0$. We introduce phases ϕ_X equal to $1/2$ if $X = C$ and 0 otherwise, in order to obtain real numbers as transformed coefficients $S^\omega_{y,X}$ [2]. This implies, if $\omega \neq 0$,

$$S^{n-\omega}_{y,X} = S^\omega_{y,X} \exp\left(2\mathrm{i}\pi(\phi_X - \phi_Y)\right) \quad . \tag{4}$$

2.2 Behaviour at the Limit

Doo and Sabin [5] have already remarked that behaviour of the limit surface at the EV is related to the eigenstructure of the subdivision matrix. Hence we refer to the eigenstructure for a scheme derived from the stencils previously described, and the most recently proposed conditions of eigenvalues and eigenvectors for convenient behaviour of the limit surface.

Eigenstructure. Let us define the matrices $\underline{S^\omega}$ with $\omega \in \{0, \ldots, n-1\}$, whose components are $S^\omega_{y,X}$ defined in Sect. 2.1 with $(Y,X) \in \{\{A\} \cup \mathscr{X}\}^2$ if $\omega = 0$ and $(Y,X) \in \mathscr{X}^2$ otherwise. Let λ^ω_p be the p-th greatest eigenvalue of $\underline{S^\omega}$ and $v^\omega_p := \left[X^\omega_p\right]_X$ the associated eigenvector which components are X^ω_p.

From (4), we deduce that, if $\omega \neq 0$, then λ^ω_p is an eigenvalue of $\underline{S^{n-\omega}}$ and $\left[X^\omega_p \exp(-2\mathrm{i}\pi\phi_X)\right]_X$ is its associated eigenvector.

Let $\underline{s}$ be the subdivision matrix with a block lower triangular structure described by Sabin [20] ($\underline{s}$ multiplies vectors which entries are associated to vertices sorted by rings and not by sectors). Using inverse discrete Fourier transform techniques, we obtain:

- λ^ω_p is an eigenvalue of $\underline{s}$;
- $\left[A^0_p,\ B^0_p \ldots B^0_p,\ C^0_p \ldots C^0_p,\ D^0_p \ldots D^0_p\right]^t$
  $\left[0,\ B^\omega_p c^{\omega,0}_{\phi_B} \ldots B^\omega_p c^{\omega,n-1}_{\phi_B},\ \ldots\right]^t$ and
  $\left[0,\ B^\omega_p s^{\omega,0}_{\phi_B} \ldots B^\omega_p s^{\omega,n-1}_{\phi_B},\ \ldots\right]^t$ are eigenvectors of $\underline{s}$

where

$$s_{\phi_X}^{\omega,k} = \sin\left(\frac{2\pi\omega}{n}(k+\phi_X)\right).$$

In particular, they define the three quadratic natural configurations [20] also called characteristic meshes [1] which vertices coordinates are (for $X \in \mathscr{X}$):

- for the cup-shape,
  A: $(0,\ 0,\ A_2^0)$, X_j: $(X_1^1 c_{\phi_X}^{1,j},\ X_1^1 s_{\phi_X}^{1,j},\ X_2^0)$;
- for the first saddle-shape,
  A: $(0,\ 0,\ 0)$, X_j: $(X_1^1 c_{\phi_X}^{1,j},\ X_1^1 s_{\phi_X}^{1,j},\ X_1^2 c_{\phi_X}^{2,j})$;
- for the second saddle-shape,
  A: $(0,\ 0,\ 0)$, X_j: $(X_1^1 c_{\phi_X}^{1,j},\ X_1^1 s_{\phi_X}^{1,j},\ X_1^2 s_{\phi_X}^{2,j})$.

Expected Eigenvalues. Conditions on eigenvalues of a convenient limit surface at the EV have been summed up by Peters Reif and Karčiauskas [15, 11]. If $\lambda := \lambda_1^1$, $\mu := \max(\lambda_1^2, \lambda_2^0)$, $\mathscr{L} := \left\{\lambda_1^0, \lambda_1^1, \lambda_1^{n-1}, \lambda_1^2, \lambda_1^{n-2}, \lambda_2^0\right\}$ and ν the greatest eigenvalue which does not belong to $\mathscr{L}$, then

Continuity

$\lambda_1^0 = 1$: necessary for convergence;
$1 > \lambda > \mu$: necessary for C^1-continuity;
$\mu = \lambda_1^2 = \lambda_2^0 > \nu$: necessary for C^2-continuity .

Let us assume that $\lambda_1^0 = 1 > \lambda > \mu > \nu$, then

Flexibility

$\mu = \lambda_1^2 > \lambda_2^0$: EV hyperbolic in sign and not L^2-elliptic;
$\mu = \lambda_2^0 > \lambda_1^2$: EV not L^2-hyperbolic;
$\mu = \lambda_2^0 = \lambda_1^2$: the scheme can generate both elliptic and hyperbolic shapes;

Principal Curvatures

$\mu < \lambda^2$: the principal curvatures converge to 0;
$\mu > \lambda^2$: at least one principal curvatures diverges;
$\mu = \lambda^2$: both principal curvatures are bounded and at least one of them does not converge to 0;

Speed of Convergence

λ is close to $1/2$: otherwise the convergence may be highly non-uniform.

Expected Eigenvectors. The only condition of eigenvectors taken into account in any tuning method is the necessary condition for scheme convergence:

$$v_1^0 := [\,1 \ldots 1\,]^t.$$

Indeed, eigenvectors have been studied less than eigenvalues, probably because it is not straightforward to translate the conditions of eigenfunctions for C^2-continuity onto eigenvectors [16, 25]. Nevertheless Gérot et al. [6] proposed conditions of eigenvectors deduced from the interpretation of subdivided meshes as a sequence of sample C^p-surfaces. Such conditions have been used by Barthe and Kobbelt [3]; they are equivalent to force the vertices of the characteristic meshes, defined in Sect. 2.2, to be samples of nice quadratic shapes:

- $X_1^1, \forall X \in \mathscr{X}$ are such that the projection of the characteristic meshes on the plane $z = 0$ are "well-shaped" planar meshes;
- the altitudes satisfy, $\forall X \in \mathscr{X}$

$$X_2^0 - A_2^0 = \tau (X_1^1)^2,$$
$$X_1^2 = \tau' (X_1^1)^2.$$

where τ and τ' are constants.

3 Designing a Scheme from an Eigenstructure

The method we propose for building the subdivision matrix from an expected eigenstructure is similar to the one used by Prautzsch and Umlauf [17]. Prautzsch and Umlauf kept the original eigenvectors and changed only the eigenvalues of the subdivision matrix in order to create a G^2-flat point at the EV. We define the small matrices $\underline{S^\omega}$ with $\omega \in \{0, \ldots, n-1\}$ (Sect. 2.2) from eigenvalues and eigenvectors which have both been changed in order to fulfil the most recently proposed expected properties (Sect. 2):

$$\underline{S^\omega} = \underline{V^\omega \Lambda^\omega} (\underline{V^\omega})^{-1}$$

where $\underline{\Lambda^\omega} = \mathrm{diag}(\lambda_1^\omega, \lambda_2^\omega, \lambda_3^\omega)$ if $\omega \neq 0$, $\underline{\Lambda^0} = \mathrm{diag}(\lambda_1^0, \lambda_2^0, \lambda_3^0, \lambda_4^0)$ otherwise, and the columns of $\underline{V^\omega}$ are the eigenvectors v_p^ω with $p \in \{1, \ldots, 3\}$ or $p \in \{1, \ldots, 4\}$ respectively. Then, the coefficients of the stencils are deduced from (2 - 3).

Notice that the stencils defined in (1) assume that the first-ring neighbours of the EV are all ordinary vertices. Moreover these stencils may be much larger than the ordinary ones, up to the complete two-ring neighbourhood of the EV (Fig. 2).

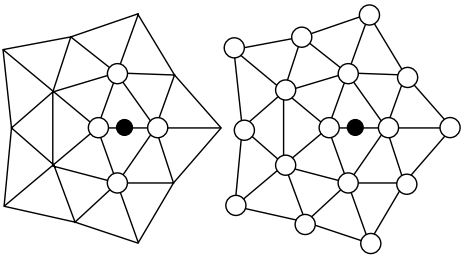


Fig. 2. Old vertices (white) involved in the stencil defining a new one (black) in the vicinity of an EV for Loop's original scheme (left) and for our tuning (right)

Furthermore, the conditions derived from expected properties do not provide a value for all eigenvalues and eigenvectors: heuristics must be used. We have tested several heuristics, such as decreasing eigenvalues and simple eigenvectors, without major changes in the resulting surface. Hence we recommend the following simple one: keeping the eigenvalues and eigenvectors corresponding to $\omega \leq 3$ as close as possible to those defined for the ordinary rules, and eigenvalues for $\omega > 3$ equal to zero. We apply it to Loop's scheme in the following section.

3.1 Loop's Scheme

In this section, we write the eigenstructure of the ordinary Loop's scheme matrices $\underline{\widehat{S}^{\omega}}$ for the successive values of ω. Of course, the matrices $\underline{\widehat{S}^{\omega}}$ satisfy all the expected conditions of eigenvalues and eigenvectors presented in Sect. 2.2. Hence we keep them as matrices $\underline{S^{\omega}}$ for the extraordinary rules in so far as they do not contradict the eigenstructure presented in Sect. 2.2.

• $\omega = \mathbf{0}$ The eigenstructure for the ordinary Loop's scheme is

$$\underline{\widehat{\Lambda}^0} = \mathrm{diag}(1, 1/4, 1/8, 1/16),$$
$$\underline{\widehat{V}^0} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 5 & 1 & 0 \\ 1 & 7 & 2 & 1 \end{bmatrix}$$

In particular the eigenvector v_2^0 satisfies (with X_1^1 coming from $\omega = 1$ and for all $X \in \mathscr{X}$):

$$A_2^0 = -1;\ X_2^0 - A_2^0 = 2(X_1^1)^2.$$

For the extraordinary rules, we choose the same eigenstructure except for $A_2^0 := \alpha$ that is kept as a free parameter which will be fixed in Sect. 4.1.

$$\underline{S^0} = \frac{1}{8}\begin{bmatrix} 8+3\alpha & -3\alpha & 0 & 0 \\ 6+3\alpha & 2-3\alpha & 0 & 0 \\ 4+3\alpha & 3-3\alpha & 1 & 0 \\ (7+6\alpha)/2 & 3-3\alpha & 1 & 1/2 \end{bmatrix}$$

• $\omega = \mathbf{1}$ The eigenstructure for the ordinary Loop's scheme is

$$\underline{\widehat{\Lambda}^1} = \mathrm{diag}(1/2, 1/8, 1/16),$$
$$\underline{\widehat{V}^1} = \begin{bmatrix} 1 & 0 & 0 \\ \sqrt{3} & 1 & 0 \\ 2 & \sqrt{3} & 1 \end{bmatrix}$$

For the extraordinary rules, we choose the same eigenstructure except for valence $n = 3$. Indeed, from Sect. 2.2 we know that $\underline{S^1}$ and $\underline{S^{3-1}}$ have the same eigenvalues. Moreover $(\lambda_1^1)^2$ must be an eigenvalue of $\underline{S^2}$ and $\left(X_p^2 \exp(-2i\pi\phi_X)\right)_X$ is an eigenvector of $\underline{S^1}$ (Sect. 2.2). Hence we choose

$$\underline{\Lambda^1} = \mathrm{diag}(1/2, 1/4, 1/8),$$
$$\underline{V^1} = \begin{bmatrix} 1 & 1 & 0 \\ \sqrt{3} & -3 & 0 \\ 2 & 4 & 1 \end{bmatrix}$$

As a consequence, if $n = 3$,

$$\underline{S^1} = \frac{1}{4(1+\sqrt{3})}\begin{bmatrix} 2\sqrt{3}+1 & 1/\sqrt{3} & 0 \\ 3 & 2+\sqrt{3} & 0 \\ 3\sqrt{3}+2 & 1/\sqrt{3} & (1+\sqrt{3})/2 \end{bmatrix}$$

and otherwise,

$$\underline{S^1} = \underline{\widehat{S}^1} = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 \\ 3\sqrt{3}/4 & 1/4 & 0 \\ 11/8 & \sqrt{3}/8 & 1/8 \end{bmatrix}$$

• $\omega = \mathbf{2}$ The eigenstructure for the ordinary Loop's scheme is

$$\underline{\widehat{\Lambda}^2} = \mathrm{diag}(1/4, 1/8, 1/16),$$
$$\underline{\widehat{V}^2} = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 4 & 1 & 1 \end{bmatrix}$$

For the extraordinary rules, we choose the same eigenstructure except for valence $n = 3$: $\underline{S^2}$ is not involved in the stencil (1), and for valence $n = 4$: $\underline{S^2} = \underline{S^{4-2}}$. Hence, for $n = 4$ we choose

$$\underline{\Lambda^2} = \mathrm{diag}(1/4, 1/4, 1/8),$$
$$\underline{V^2} = \begin{bmatrix} 1 & 1 & 0 \\ 3 & -3 & 0 \\ 4 & 4 & 1 \end{bmatrix}$$

As a consequence, if $n = 4$,

$$\underline{S^2} = \frac{1}{4}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1/2 \end{bmatrix}$$

and otherwise,

$$\underline{S^2} = \underline{\widehat{S}^2} = \frac{1}{4}\begin{bmatrix} 1 & 0 & 0 \\ 3/2 & 1/2 & 0 \\ 9/4 & 1/4 & 1/4 \end{bmatrix}$$

• $\omega = \mathbf{3}$ The eigenstructure for the ordinary Loop's scheme is

$$\underline{\widehat{\Lambda}^3} = \mathrm{diag}(1/8, 1/8, 1/16),$$
$$\underline{\widehat{V}^3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 8 & 0 & 1 \end{bmatrix}$$

Notice that because $\underline{\widehat{S}^3} = \underline{\widehat{S}^{6-3}}$, $\omega = 3$ is the particular case for $n = 6$. Concerning the extraordinary rules, no eigenstructure must be defined for $n < 6$. Aiming for the most natural generalisation, we propose for $n > 6$ the following eigenstructure:

$$\underline{\Lambda^3} = \mathrm{diag}(1/8, 1/16, 1/32),$$
$$\underline{V^3} = \begin{bmatrix} 1 & 0 & 0 \\ 3\sqrt{3} & 1 & 0 \\ 8 & 1 & 1 \end{bmatrix}$$

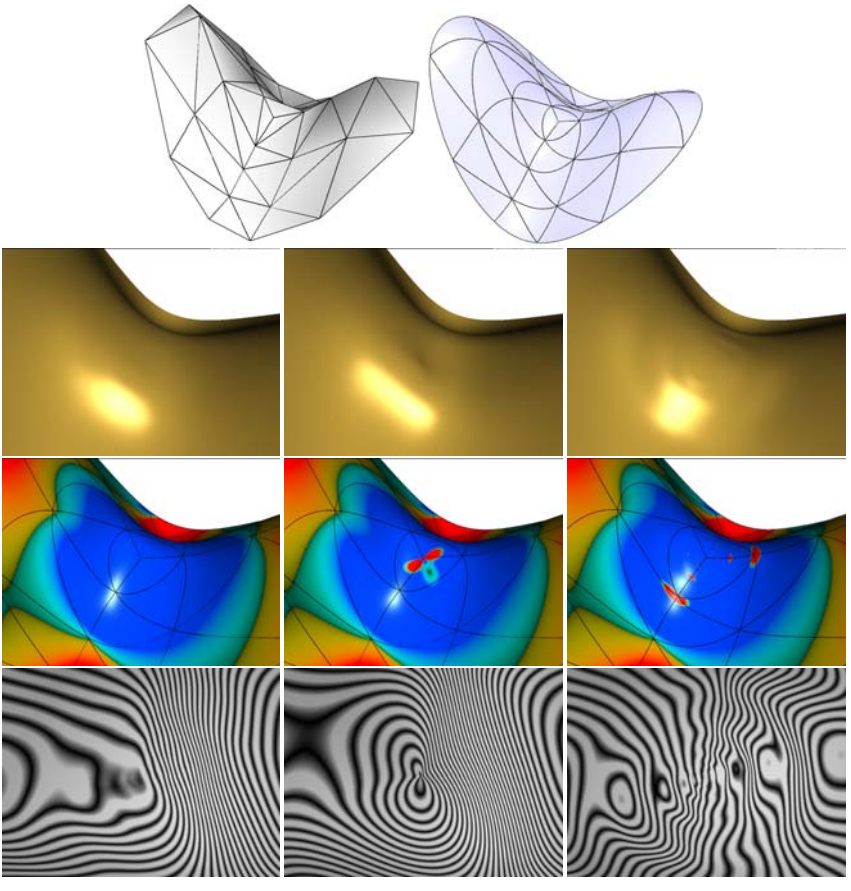


Fig. 3. Concentric waves generated with purely eigenstructure-based rules shade off when a blending smoothes the antagonism between EV and its regular neighbourhood. First line: the original mesh and the surface produced after 6 subdivision steps. The following nine images represent (from top to bottom respectively) the enlighted surface, mapped Gaussian curvature, and reflection lines for (from left to right respectively) surfaces subdivided with blended rules, with original Loop's rules, and with pure eigenstructure-based rules.

As a consequence, if $n > 6$,

$$\underline{S^3} = \frac{1}{8}\begin{bmatrix} 1 & 0 & 0 \\ (3\sqrt{3})/2 & 1/2 & 0 \\ (24-3\sqrt{3})/4 & 1/4 & 1/4 \end{bmatrix}$$

• $\omega > 3$ We propose $\underline{S^\omega} = 0$.

With these matrices $\underline{S^\omega}$, $\alpha = -1$ as in the original scheme, and (2–3) the stencils for vertices around an EV are completely defined. They define a tuning of Loop's scheme exclusively focused on properties in the limit at the EV. Figure 3 illustrates side effects

such an exclusiveness may cause on the surface around the EV. These side effects are discussed in the following section.

3.2 Antagonism between EV and Its Neighbourhood Creates Concentric Waves

As illustrated in Fig. 3, the surfaces generated by these extraordinary stencils based on expected eigenstructure of the subdivision matrix only, are far from satisfactory. Two reasons can explain this misbehaviour.

Firstly, (11) implies that new vertices belonging to the two-first-ring around the EV must be defined with extraordinary stencils. But they belong to the neighbourhood of ordinary vertices as well. As a consequence, their definition should follow ordinary rules at the same time.

Secondly, the regular part of the initial mesh defines a Box-Spline surface with an n-sided hole around each EV. If an extraordinary rule is defined only for the new EV, then the meshes through subdivision are all control polyhedra of this same Box-Spline surface. On the other hand, if extraordinary rules are also used for other vertices in the vicinity of the EV, then the first ring of Box-Spline patches surrounding the n-sided hole is modified. And if these rules are defined for convenient limit surface at the EV (as the ones proposed in the previous section), nothing ensures that the deformation they induce on surrounding Box-Spline rings is consistent with the shape of the global surface.

Intuitively, a compromise must be found between a nice limit surface at the EV and the side effects created by the uncontrolled transition from EV to ordinary vertices. We propose the definition of a new vertex around an EV as a convex combination of the ideal new vertex from the EV point of view (defined with the rules given in the previous section) and the ideal new vertex from the ordinary vertices' point of view (defined with ordinary rules).

4 Blending the Stencils for Smoothing the Antagonism

Let y_j be the new vertex defined with eigenstructure-based rules (11). Let $\hat{y}_j$ be the new vertex defined with ordinary rules. Let $\tilde{y}_j$ be the final new vertex resulting from blending the extraordinary and ordinary stencils:

$$\tilde{y}_j := \sigma_y y_j + (1 - \sigma_y)\hat{y}_j$$

with

$$\sigma_y := \sigma(Y_1^1)$$

where Y_1^1 is the radius of ring Y in the projection of characteristic meshes onto the plane $z = 0$ (see Sect. 2.2 and Fig. 4), and $\sigma(l)$ is a cubic polynomial interpolating Hermite conditions $(\sigma(0) = 1, \sigma'(0) = 0)$ and $(\sigma(\Delta) = 0, \sigma'(\Delta) = 0)$:

$$\sigma(l) := 2\frac{l^3}{\Delta^3} - 3\frac{l^2}{\Delta^2} + 1.$$

The symbol Δ defines the end of the influence of the eigenstructure-based rules in the definition of new vertices. Notice that this blending process do not enlarge the stencils (11) defined from the expected eigenstructure.

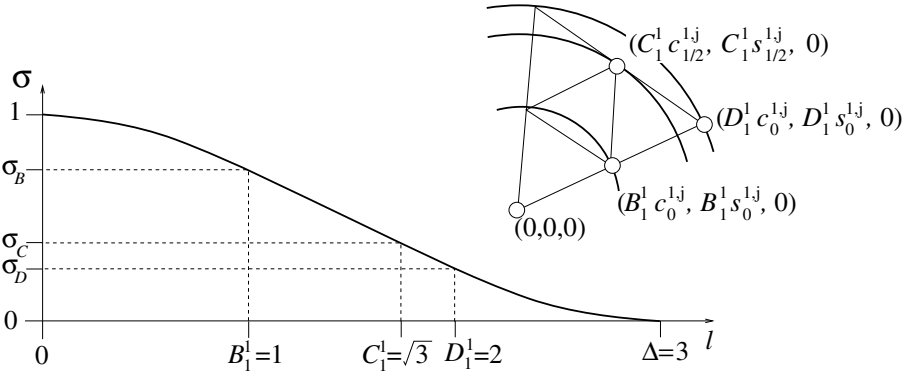


Fig. 4. Weights of the eigenstructure-based rules in the definition of new vertices, and rings in the projection of characteristic meshes onto the plane $z = 0$

4.1 Loop's Scheme

For Loop's scheme, we have chosen in Sect. 3.1 $v_1^1 := \begin{bmatrix} 1 & \sqrt{3} & 2 \end{bmatrix}^t$. We choose $\Delta = 3$ in order to include the ring D which radius is equal to 2 (see Fig. 4). Thus, the weights of the eigenstructure-based rules in the definition of new vertices are

$$\sigma_A = \sigma(0) = 1, \quad \sigma_B = \sigma(1) = \tfrac{20}{27},$$
$$\sigma_C = \sigma(\sqrt{3}) = \tfrac{6\sqrt{3}}{27}, \; \sigma_D = \sigma(2) = \tfrac{7}{27}.$$

This blended scheme needs to be eigenanalysed. The blending and discrete Fourier transform being linear, if $\underline{\hat{S}^\omega}$ are small matrices for ordinary rules then the small matrices $\underline{\tilde{S}^\omega}$ for the blended scheme are defined by

$$\tilde{S}^\omega_{y,X} = \sigma_Y S^\omega_{y,X} + (1 - \sigma_Y)\hat{S}^\omega_{y,X}.$$

Let us remember that in Sect. 3.1, we left as free parameter $A_2^0 := \alpha$. We obtain the following eigenvalues:

$$\lambda_1^1 = \begin{cases} \frac{53}{144} - \frac{5\sqrt{3}}{108} + \frac{1}{432}\sqrt{19641 - 9240\sqrt{3}} & \text{if } n = 3 \\ \frac{101}{216} + \frac{7}{108}c_0^{1,1} & \text{if } n > 3 \end{cases}$$
$$\lambda_1^2 = \begin{cases} \frac{47}{216} & \text{if } n = 4 \\ \frac{61}{216} + \frac{7}{108}c_0^{2,1} & \text{if } n > 4 \end{cases}$$
$$\lambda_2^0 = \frac{7}{72}\alpha + \frac{25}{72}$$

Here, α is chosen in order to obtain $\lambda_2^0 = \left(\lambda_1^1\right)^2$. Among the expected eigenvalues we do not satisfy the non-polar artifact condition anymore ($\lambda \neq 1/2$), we still fulfil the necessary condition for C^1-continuity ($1 > \lambda > \mu$) but not for C^2-continuity ($\lambda_1^2 \neq \lambda_2^0$). If the valence of the EV is in $\{4,5\}$, the EV is not L^2-hyperbolic and both principal curvatures are bounded and at least one of them does not converge to 0 ($\mu = \lambda_2^0 = \left(\lambda_1^1\right)^2 > \lambda_1^2$). If the valence of the EV is greater or equal to 7, the EV is hyperbolic

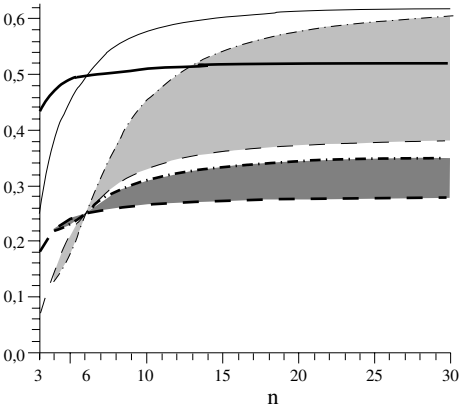


Fig. 5. Eigenvalues for the original Loop's scheme (thin lines) and for our blended scheme (thick lines) for different valences n for the EV. Solid line for λ_1^1, dash line for λ_0^2 and dash-and-dot line for λ_2^1. The solid line should be close to 0.5 and the grey area highlights the difference between $\lambda_0^2 = (\lambda_1^1)^2$ and λ_2^1 which takes the scheme away from necessary condition for C^2-convergence: this area is smaller and the solid line is closer to 0.5 for our blended scheme.

in sign and not L^2-elliptic and at least one of the principal curvatures diverges ($\lambda_2^0 = (\lambda_1^1)^2 < \mu = \lambda_1^2$). However the eigenvalues remain quite close to these conditions, which explains the visually good shapes obtained. In particular they are better than the original Loop's scheme eigenvalues (see Fig. 5), without the concentric wave pattern caused by the perfect eigenvalues schemes (see Fig. 3).

Figure 6 illustrates the eigensurfaces associated with the eigenvectors of the blended scheme, while Fig. 7 shows its basis functions for various valences.

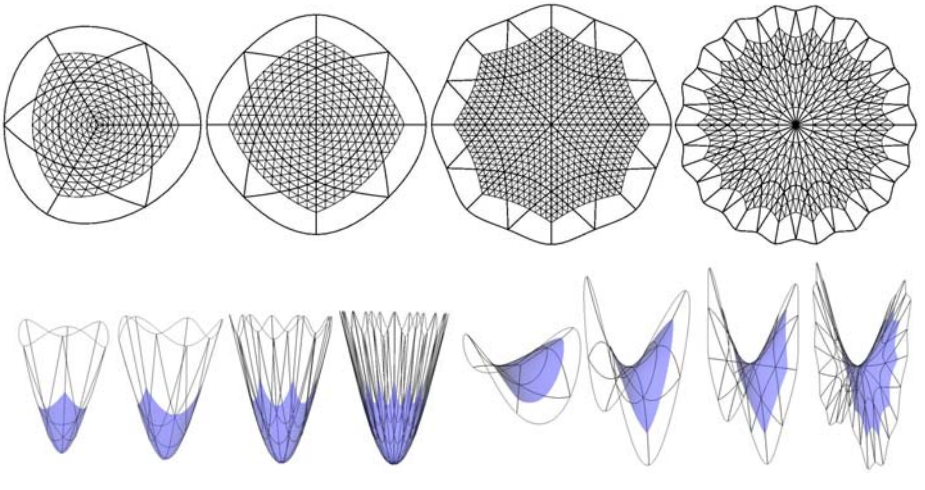


Fig. 6. Characteristic maps and meshes of the blended scheme, for valences 3, 4, 8 and 18. The injectivity of maps is required for regularity of the surface at extraordinary vertices.

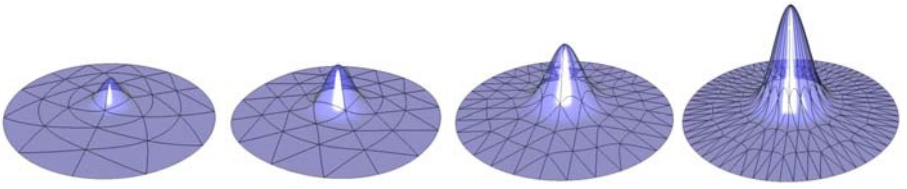


Fig. 7. Basis functions of our blended scheme, for valences 3, 4, 9 and 18

5 Implementation Issues

The subdivision is performed in two steps: after subdividing the regular parts, the neighbourhood of each EV is computed. As explained in Sect. 2.1, the stencil coefficients $s^k_{y,X}$ can be easily determined from matrices $\underline{S^\omega}$ with real entries. They can be computed and stored in a 4-dimensional array S where $S[Y][X][n][k]$ is the weight $s^k_{y,X}$ of $X_{j\pm k}$ in the definition of y_j around an EV whose valence is equal to n. At each step of subdivision, we store each extraordinary vertex topological neighbourhood in a list, increasing the efficiency of the process. Notice that since no EV is generally added through subdivision, the local treatment of EV neighbourhood is constant in time.

6 Results

Instead of a limit surface analysis [22], [1], we propose to evaluate the global surfaces behaviour generated by our scheme through the shape and the curvature distribution analysis in the early subdivision steps (near 6). We have intensively tested each scheme on various geometric models. In Fig. 9 and Fig. 10, some of the most relevant ones containing one EV surrounded by ordinary vertices are shown. Figure 8 and Fig. 11 illustrate some results on complete meshes.

6.1 Curvature Distribution Rendering

We use the surface's piecewise quartic parametrisation to compute the Gaussian and mean curvatures of each regular vertex. We subdivide until the central hole at the extraordinary vertex is smaller than one pixel. For each image the measures are thresholded between $[-1,1]$, the hue H (from HSV coding) is uniformly distributed from cyan to blue for the interval $[-1,0]$, and from yellow to red in $[0,1]$. In addition we normalise the value V (from HSV coding) with norms of the measures to emphasis the sign changes in attempt to see in the early subdivision steps the so-called hybrid shapes, which can lead to curvature oscillations [11].

6.2 Other Tunings of Loop's Scheme

In order to evaluate the global surface behaviour generated by our scheme, we compare our scheme to stationary standard-tuned Loop's schemes on several control models.

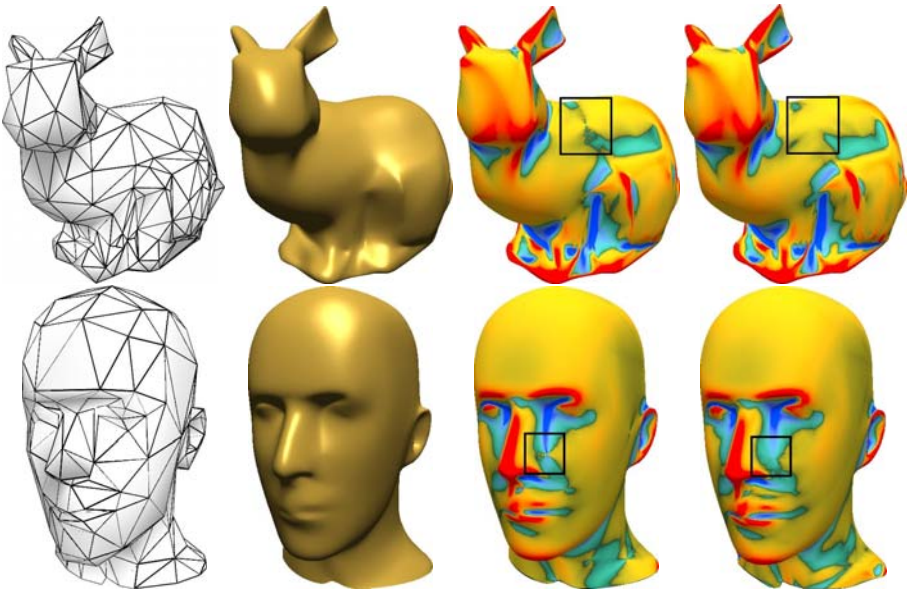


Fig. 8. From top to bottom: Two complete meshes with several EVs. **From left to right:** control mesh (a), surface (b), mean curvature mapped on surfaces generated after 6 steps with original Loop's scheme (c) and our blended scheme (d).

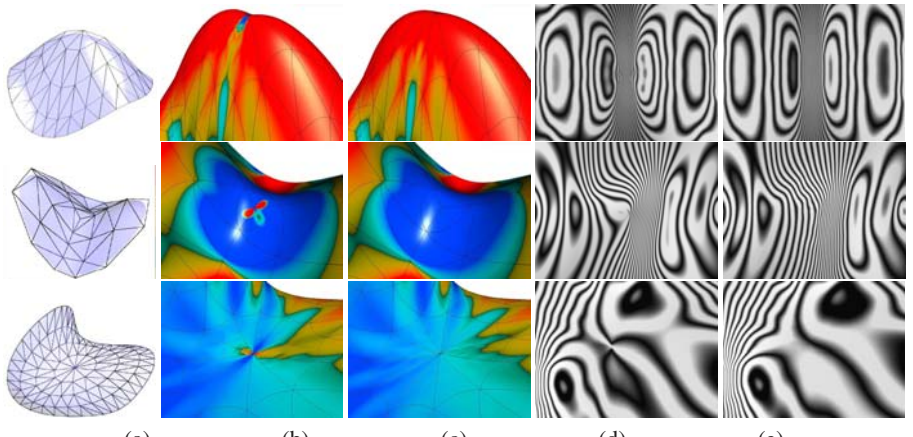


Fig. 9. From top to bottom: different geometric models with one extraordinary valence and a mapped curvature (g stands for Gaussian and m for mean curvature) crest (9,g), low-valence saddle (3,g), high-valence hill (12,m). **From left to right:** control mesh (a), curvature (b,c) and reflection lines (d,e) mapped on surfaces generated after 6 steps with original Loop's scheme (b,d) and our blended scheme (c,e).

Among them, we choose the scheme originally proposed by Loop [13], the two schemes proposed by Barthe and Kobbelt's [3] to improve the curvature or to reduce polar distortions, the G^2-scheme presented by Prautzsch and Umlauf [17], and the scheme proposed by Karčiauskas, Peters and Reif tuned to obtain $\lambda_1^2 = \lambda_1^0$ for valence $n > 3$ [11]. Notice that the tuning method proposed by Augsdörfer et al. [1] does not fit well on Loop's scheme. It first requires the reduction of the number of freedoms by imposing $\mu = \lambda^2$, before optimising λ for the best curvature behaviour around the EV. However, in the case of Loop's scheme, $\mu = \lambda^2$ defines the value of λ.

6.3 Curvature Distribution and Fairness Improvement

Figure 9 shows that the classical Loop's scheme is not appropriate for some geometric models. It yields undesirable behaviour, e.g. curvature oscillations and major shape distortions. For saddle-like meshes, Loop's scheme produces an acute angle as the valence increases, which can be easily seen on the reflection lines. For the ridge-like meshes, it leads to a saddle-like behaviour, as do all the classic directional box-spline deriving schemes [15].

Our scheme corrects some of these artifacts (see Fig. 9). The fairness and the curvature general distribution, even for the valence 3, are greatly improved. In comparison with other tuning methods, it achieves the best global shapes (see Fig. 10).

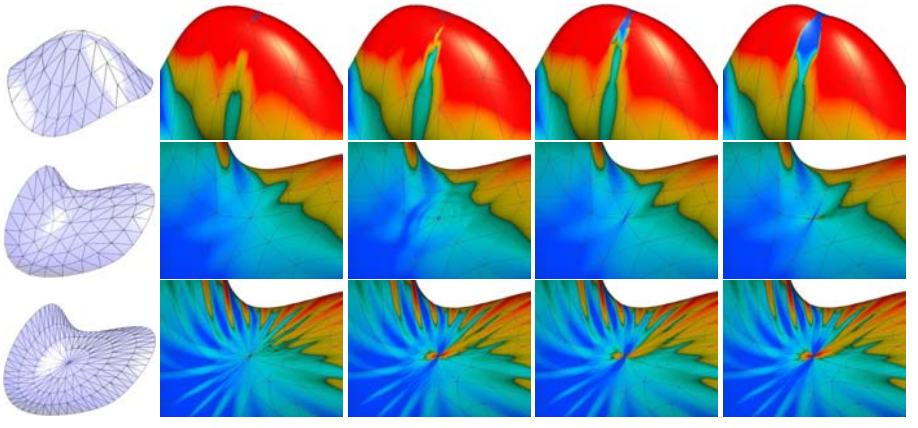


Fig. 10. From top to bottom: different geometric models with one extraordinary valence and a mapped curvature (g stands for Gaussian and m for mean curvature): crest (8,g), hill (8,m) and high-valence hill (18,m). **From left to right:** control mesh (a), and curvature mapped on surfaces generated after 6 steps with different tuned Loop's schemes: (b) our blended scheme, (c) Barthe and Kobbelt's reducing the polar artifact (first line), Barthe and Kobbelt's optimized to improve the curvature (second line) [3], or Warren and Weimer's [24] (third line), (d) Karčiauskas, Peters and Reif's [11], (e) Prautzsch and Umlauf's [17].

7 Limitations

As already noted in Sect. 3, our blended scheme uses much wider stencils than the other schemes. This has little bearing on computing efficiency (see Sect. 5), but it prevents the use of our scheme in the first subdivision step if one EV is in the one-ring neighbourhood of another EV: Equation (1) could not be written. Moreover, our scheme defines extraordinary stencils for the EV and its two-ring neighbourhood. As a consequence, if one EV is in the two-ring neighbourhood of another EV, then different extraordinary stencils (and not only one) can be in competition with an ordinary stencil. In these cases, another scheme with smaller stencils or heuristics convenient for the first subdivision steps must be used at the two first steps.

Notice that our tuning is designed in order to smooth some artifacts localised at the EV which are produced by the classical Loop's scheme after several subdivision steps, while erasing the concentric wave pattern which normally occurs when extraordinary stencils are defined for more vertices than the EV. As a consequence, if another scheme is used at the very first steps only, some artifacts localised at the EV together with the concentric wave pattern still shade off if our scheme is used in the following steps (see Fig. 8).

But it also means that our tuning method is not designed to remove other artifacts which are created in the first subdivision steps, such as the rotational artifact on a prism-like model (see Fig. 11). Only a non-stationary adaptation of Loop's scheme could obtain the flexibility required for an acceptable modelling of all these shape definitions.

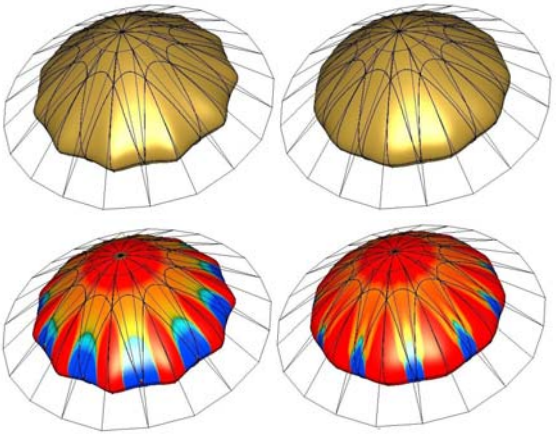


Fig. 11. Prism-like model presents a challenging geometry for all the stationary schemes. Mean curvature is mapped onto the surface generated by original Loop's scheme (left) and by ours (right). Our scheme improves the surface behaviour but the rotational artifact [21] is still visible.

8 Conclusion

Recent methods for tuning a subdivision scheme, where extraordinary stencils are proposed not only for the extraordinary vertex (EV) but also for its neighbours, create a

concentric wave pattern around the EV. We have proposed that this concentric wave pattern results from contradictory behaviours for a convenient shape at the limit EV, and for convenient transition onto the surrounding regular surface.

Firstly we have shown that even a scheme which fulfils the most recently proposed conditions for good behaviour of the limit surface, may still produce this wave pattern. The designing of this scheme extends the method proposed by Prautzsch and Umlauf [17]: the subdivision matrix is built from an expected eigenstructure. Whereas Prautzsch and Umlauf kept the original eigenvectors and changed only the eigenvalues of the subdivision matrix in order to create a G^2 flat point at the EV, we change both eigenvalues and eigenvectors in order to fulfil the most recently proposed conditions for good behaviour of generated surfaces [15, 6]. In addition, we use discrete Fourier techniques to make the definition of the expected eigenstructure as simple as possible.

Then, in order to achieve a compromise between a nice limit surface at the EV and the side effects created by the uncontrolled transition from EV to ordinary vertices, we have proposed the definition of a new vertex around an EV as a convex combination of the ideal new vertex from the EV point of view (defined with the rules given previously and the ideal new vertex from the ordinary vertices point of view (defined with ordinary rules). The weight of the extraordinary rules decreases as the new vertex is topologically farther from the extraordinary element. We applied this principle to Loop's scheme. Even if all artifacts are not removed, such as the rotational artifacts, the concentric wave pattern shades off whereas the expected conditions are not spoiled too much. This tuning method remains simple and useful, involving no optimisation process.

The practical generalisation of this method to other schemes than Loop's requires extra notation and phases ϕ_X if each vertex in the vicinity of interest is not uniquely determined by a couple (ring, sector) (e.g. Catmull and Clark's [4]) or if the scheme rotates the lattice (e.g. Kobbelt's [9] or Velho's [23]). The generalisation to *dual* schemes near an extraordinary face (e.g. Doo and Sabin's [5]) should not add other difficulties once the stencils (1) are rewritten.

It seems difficult to remove the other artifacts such as the rotational artifact, while staying in the simple but useful context of stationary subdivision schemes. Other measures related more to signal processing than to differential geometry could be proposed. But in order to keep the tuning process as simple as possible in such a way that anyone can apply it easily to any scheme, these measures should not be used in an optimisation process.

References

1. Augsdörfer, U.H., Dodgson, N.A., Sabin, M.A.: Tuning subdivision by minimising Gaussian curvature variation near extraordinary vertices. Computer Graphics Forum 25(3) (2006); (Proc. Eurographics 2006)
2. Barthe, L., Gérot, C., Sabin, M.A., Kobbelt, L.: Simple computation of the eigencomponents of a subdivision matrix in the Fourier domain. In: Dodgson, N.A., Floater, M.S., Sabin, M.A. (eds.) Advances in Multiresolution for Geometric Modelling, pp. 245–257. Springer, Heidelberg (2005)
3. Barthe, L., Kobbelt, L.: Subdivision scheme tuning around extraordinary vertices. Computer Geometric Aided Design 21(6), 561–583 (2004)

4. Catmull, E., Clark, J.: Recursively generated B-spline surfaces on arbitrary topological meshes. Computer Aided Design 10(6), 350–355 (1978)
5. Doo, D., Sabin, M.A.: Behaviour of recursive division surface near extraordinary points. Computer Aided Design 10(6), 356–360 (1978)
6. Gérot, C., Barthe, L., Dodgson, N.A., Sabin, M.A.: Subdivision as a sequence of sampled cp surfaces. In: Dodgson, N.A., Sabin, M.A. (eds.) Advances in Multiresolution for Geometric Modelling. Springer, Heidelberg (2005)
7. Ginkel, I., Umlauf, G.: Controlling a subdivision tuning method. In: Cohen, A., Merrien, J.-L., Schumaker, L.L. (eds.) Curve and Surface Fitting: Avignon 2006, pp. 170–179. Nashboro Press (2007)
8. Ivrissimtzis, I., Dodgson, N.A., Sabin, M.A.: A generative classification of mesh refinement rules with lattice transformations. Computer Aided Geometric Design 21(1), 99–109 (2004)
9. Kobbelt, L.: $\sqrt{3}$-subdivision. In: SIGGRAPH 2000 Conference Proceedings, pp. 103–112 (2000)
10. Karčiauskas, K., Peters, J.: Concentric tessellation maps and curvature continuous guided surfaces. Computer Aided Geometric Design 24(2), 99–111 (2007)
11. Karčiauskas, K., Peters, J., Reif, U.: Shape characterization of subdivision surfaces–case studies. Computer Geometric Aided Design 21(6), 601–614 (2004)
12. Levin, A.: Modified subdivision surfaces with continuous curvature. Computer Graphics 25(3), 1035–1040 (2006); Proc. SIGGRAPH 2006
13. Loop, C.: Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah (1987)
14. Loop, C.: Bounded curvature triangle mesh subdivision with the convex hull property. The Visual Computer 18, 316–325 (2002)
15. Peters, J., Reif, U.: Shape characterization of subdivision surfaces: basic principles. Computer Aided Geometric Design 21(6), 585–599 (2004)
16. Prautzsch, H.: Smoothness of subdivision surfaces at extraordinary points. Adv. Comput. Math. 9, 377–389 (1998)
17. Prautzsch, H., Umlauf, G.: A g^1 and g^2 subdivision scheme for triangular nets. International Journal on Shape Modelling 6(1), 21–35 (2000)
18. Reif, U.: A unified approach to subdivision algorithm near extraordinary vertices. Computer Geometric Aided Design 12, 153–174 (1995)
19. Reif, U.: A degree estimate for subdivision surfaces of higher regularity. Proc. Amer. Math. Soc. 124(7), 2167–2174 (1996)
20. Sabin, M.A.: Eigenanalysis and artifacts of subdivision curves and surfaces. In: Iske, A., Quak, E., Floater, M.S. (eds.) Tutorials on Multiresolution in Geometric Modelling, pp. 69–97. Springer, Heidelberg (2002)
21. Sabin, M.A., Barthe, L.: Artifacts in recursive subdivision surfaces. In: Cohen, A., Merrien, J.-L., Schumaker, L.L. (eds.) Curve and Surface Fitting: Saint-Malo 2002, pp. 353–362. Nashboro Press (2003)
22. Umlauf, G.: Analysis and tuning of subdivision algorithms. In: SCCG 2005: Proceedings of the 21st spring conference on Computer graphics, pp. 33–40. ACM Press, New York (2005)
23. Velho, L., Zorin, D.: 4-8 subdivision. Computer Geometric Aided Design 18(5), 397–428 (2001)
24. Warren, J., Warren, J.D., Weimer, H.: Subdivision Methods for Geometric Design: A Constructive Approach. Morgan Kaufmann Publishers Inc., San Francisco (2001)
25. Zorin, D.: Stationary subdivision and multiresolution surface representations. PhD thesis, California Institute of Technology (1997)
26. Zorin, D., Schröder, P.: Subdivision for modeling and animation. In: SIGGRAPH 2000 course notes (2000)

Simplification of FEM-Models on Cell BE

Jon Mikkelsen Hjelmervik[1,2] and Jean-Claude Léon[2]

[1] Sintef ICT, Pb. 124 Blindern, N-0134 Oslo
jon.m.hjelmervik@sintef.no
[2] Grenoble Universités, Grenoble Institute of Technology, G-SCOP Laboratory,
46 av. Félix, Viallet, F-38000
jean-claude.leon@inpg.fr

Abstract. Preparing a CAD model for Finite Element (FE) analysis can be a time-consuming task, where shape and mesh simplifications play an important role. It is important that the simplified model has the same mechanical properties as the original one, and that the deviation from the original stays within a given tolerance.

Most FE mesh simplification algorithms are either fully or partially sequential, and are therefore not suitable for architectures with high levels of parallelism. Furthermore, the use of processors such as GPUs of IBMs Cell BE require algorithms to be adapted to benefit from their computational advantages. Here, we present an algorithm written for parallel processors, and its implementation for the Cell BE.

1 Introduction

Simplification of triangulations is often used in computer graphics, where it is used to reduce the complexity of 3D models before rendering. Often, the user specifies the target number of triangles desired. The simplification software aims at creating a triangulation with the given number of triangles that qualitatively resembles the original model.

FE Analysis (FEA) allows engineers to analyze properties like strength and heat conductivity in a simulated environment. The analysis can be performed on simple components, such as beams, or on complex models, and large assemblies such as models of entire airplanes. The components usually originate from 3D scanning or CAD systems. Even though CAD systems are integrating FEA tools into their tool chains, the process from a CAD model to a fully usable FE mesh remains a manual and time consuming process. In the automotive industry it can take up to four months to create a mesh for a car. In this text we use the terms mesh and FE mesh for shape representations adapted for simulation purposes.

CAD models use smooth shapes such as NURBS, sphere segments and torus segments to represent the surface of objects. Solid objects are represented by their boundary surfaces. Transforming their shape is often time consuming and tedious. FE meshes on the other hand are usually piecewise planar. Volume meshes, such as tetrahedral meshes, are often used to represent solid objects in FE analysis and mesh generators are powerful when the input model is locally compatible with the desired FE size. Model

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 261–273, 2010.

preparation brings the CAD model to a shape that is well suited for the meshing process. Later, the prepared model is used as a starting point for meshing. To improve and automate the model preparation, its starting point can be shifted from CAD models to triangulations, enabling also a larger range of shape changes. The domain (2D or 3D) being described by a triangulation, it is then subjected to a FE mesh generation process whose output is called the FE mesh. The target size for each finite element is dependent of the required accuracy. In addition, there may be a number of other important requirements for a mesh, including maximal/minimal angles, maximal valence and relative size of neighbor elements. These requirements are not meaningful for designers so even if a CAD model is exported into the right format, it will not necessarily be acceptable as a FE mesh. Hence, the need for a shape transformation process prior to a FE mesh generation.

During the last five years the trend in commodity hardware have gone from single-core processors to homogeneous multi-core or heterogeneous many-core processors. This evolution is driven by difficulties in developing faster cores as well as the evolution of highly parallel processors, e.g., Graphics Processing Units (GPUs).

The development in hardware has made research in new algorithms necessary, as most algorithms are unsuited for the new massively parallel architectures. During the past years the interest for such research has increased tremendously, and has grown into a new discipline in computer science called "General-Purpose Computing on GPUs" (GPGPU). Research in GPGPU has attracted the interest from researchers all over the world. Due to the increase in flexibility of GPUs more and more algorithms can be adapted to take advantage of them. The research has led to development of new algorithms as well as new uses of algorithms not commonly used. Bitonic sort by Baxter [1], which has been successfully implemented on GPUs by Purcell et al. [2], is one of many examples. For a thorough overview of the field please refer to [3].

Another trend in processor design is heterogeneous processors such as the *Cell Broadband Engine Architectures* (Cell BE) from Sony, IBM and Toshiba. The IBM Cell BE is targeting both supercomputers and the home entertainment market. In addition to one or more traditional processor core(s), called Power Processing Elements (PPEs), the Cell BE include a number of *thin* cores called Synergistic Processing Elements (SPEs). The current Cell BE consists of one PPE and eight SPEs. IBM is planning to release a updated version featuring 32 thin cores and two fat cores, yielding a total of one teraflop per second. In contrast to GPUs, the thin cores of the Cell BE operate independently, and can even execute different programs. This places the Cell BE between current GPUs and multi-core CPUs when it comes both to flexibility and performance.

In this paper we present an algorithm and its implementation on multi-core CPUs and the Cell BE. The goal of the algorithm is to yield high scalability beyond the eight cores we tested it for on the Cell BE. We had the 32 core Cell BE and GPUs in mind when developing the algorithm.

2 Related Work

Computer models of product components often contain numerous details that are part of the component shape "as-manufactured". To meet the objectives and hypotheses of

component behavior simulation and reduce the time spent in the FEA process, it is advantageous that these details are removed. Shape simplification is important in computer graphics as well as in preparation of models for FEA. The wide range of applications has led to many different approaches to provide transformation operators, each useful for its applications. Gotsman et al. [4] give a good overview of the algorithms, and we will here describe some of the key features of selected algorithms and data structures.

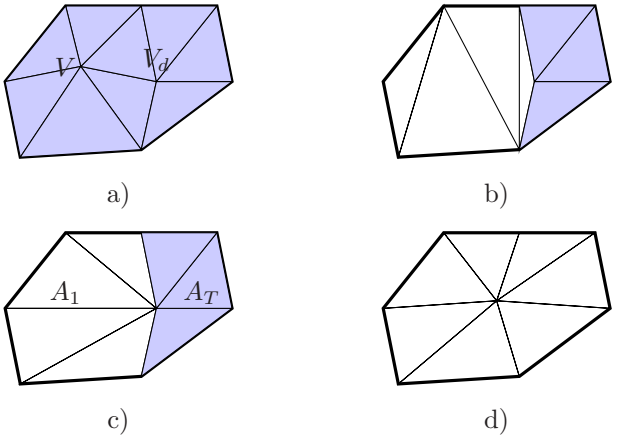


Fig. 1. Vertex removal operations, unchanged areas are marked in blue. a) original triangulation, b) arbitrary re-triangulation, c) half-edge collapse and d) edge collapse.

Most simplification algorithms are based on removing one vertex at a time, possibly grouping removals into passes. Before a vertex is removed, a number of decimation criteria are tested. The most common decimation criteria either restrict the geometric deviation between the input and simplified models, or verify that the topology of the surface is maintained. Figure 1 illustrates commonly used schemes for removing a single vertex V. Figure 1b) depicts a general vertex removal where the neighborhood surrounding the vertex V is remeshed freely. One example of this scheme is Schroeder et al. [5], where the remeshing is determined mainly by feature edges and aspect ratios. Hoppe [6] used edge collapse, where one edge is collapsed, placing its two vertices at the same position, leaving two triangles degenerated. This operation is illustrated in Figure 1d).

Kobbelt et al. [7] introduced half-edge collapse, illustrated in Figure 1c), where an edge is collapsed by moving *one* of the vertices V onto the other V_d. The main difference between edge collapse and half-edge collapse is that the latter has a smaller affected area A_1 and keeps the original vertex locations. As with edge collapse, half-edge collapse can be extended to support non-manifold surfaces.

Shape simplification can be performed in parallel on shared memory parallel architectures, providing the vertices being removed at the same time are not too close to each other. A common strategy is to create a set of vertices that can be removed simultaneously without one vertex removal being influenced by any of the others. Such

independent sets can be created in parallel. Dadoun and Kirkpatric [8] and Karp and Wigderson [9] presented algorithms that find independent set in polylogarithmic time (assuming a very high number of processors).

The main challenge for parallel algorithms is to find "good" independent sets in parallel. In a sequential setting, it is common to use greedy algorithms that use a cost function to guide the selection of vertices, and a "good" independent set is one where the decimation cost is low for all the vertices. Franc and Scala [10] proposed an algorithm for finding such a set without sorting the vertices, however their method for finding the independent set is sequential.

Botsch et al. [11], proposed to improve performance of evaluation of decimation criteria by using the GPU. In their work, the decimation criterion is expressed as threshold on the distance from the decimated surface to the original. The criterion is checked by sampling a piecewise linear approximation to the signed distance field attached to the original surface, and comparing the sampled value to the tolerance. In their work, the approximation to the signed distance field is computed using a CPU-based implementation of fast marching methods. This is transferred to a 3D texture in graphics memory. Then, the triangles that are to be checked are rendered using this texture. This is a first contribution to an efficiency increase in a decimation operator through the use of heterogeneous processor architectures.

Hjelmervik and Léon [12] presented an hybrid GPU-CPU simplification algorithm. In their work, the GPU performs the most computationally intensive tasks, while the CPU maintains the data structure. This means that data must be sent between the system memory and graphics memory, which imposes overhead. The overhead due to memory transfers makes this algorithm only feasible when decimation criteria with high arithmetic intensities are involved.

DeCoro and Tatarchuk [13] presented the first algorithm to perform the entire simplification on the GPU. They implemented vertex clustering using the geometry shader to discard degenerated triangles. Since the GPUs support *stream out* of vertex data, the simplified version can be reused for further simplification.

Simplification algorithms often sort the vertices based on their decimation cost. Sorting is a fundamental building block for a wide variety of applications. Parallel sorting algorithms suitable for heterogeneous architectures have therefore attracted the attention from several research groups. GPUTeraSort by Govindaraju et al. [14] is successfully implemented on multi-core processors and GPUs, and AA-sort by Inoue et al. [15] has shown high speedups for multi-core processors and the Cell BE processor.

3 Description of the Algorithm

Our main goal is to develop a flexible algorithm for shape simplification that is suitable for many-core heterogeneous architectures. To take advantage of the high level of parallelism, it is important that a large number of vertices are removed simultaneously and that all steps in the algorithm can be performed in a parallel fashion. Existing thin cores do not have the capability to allocate or deallocate system memory, prohibiting implementation of algorithms that use dynamic memory.

The demand for performance improvement seems most crucial either when considering simplification of large models where computationally demanding decimation

criteria are in play and/or when the simplification must be performed at an interactive rate. However, many applications use QEM or other computationally inexpensive decimation criteria. Performance may also be important for these applications, either to improve the interactivity or to simplify assemblies consisting of a large number of less detailed objects. Thus, low overhead is a key feature of our algorithm. Furthermore, the algorithm should be extendable to transfer properties attached to the faces (or edges or vertices) of the triangulation to enable the implementation of mechanical criteria like the transformation of pressure fields.

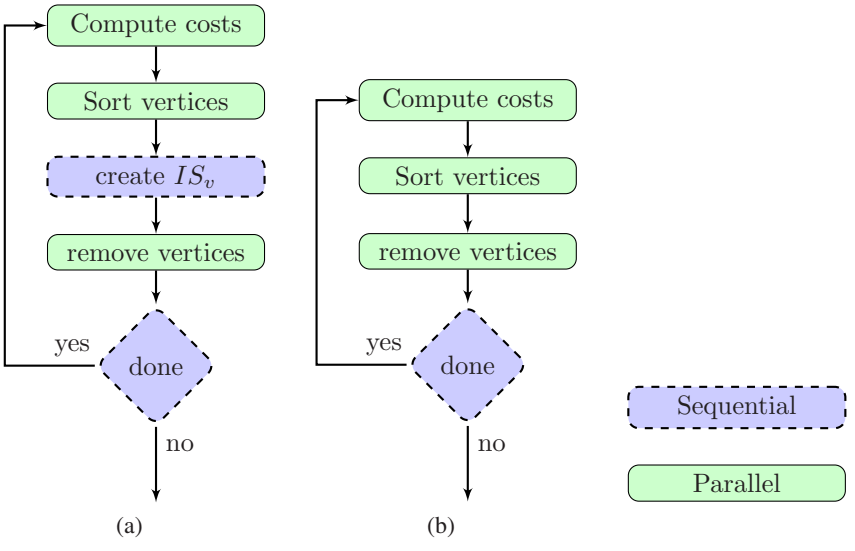


Fig. 2. Flowchart of simplification algorithms. Figure a) illustrates a traditional simplification algorithm. By removing the vertices as they are inserted into IS_v we get the flowchart illustrated in b).

A simplified view of popular simplification algorithms is illustrated in Figure 2(a). Typically, an independent set of vertices, IS_v is created using a greedy, sequential algorithm, which inserts vertices one by one into IS_v. Such a set is used to ensure that no neighboring vertices are removed simultaneously. Indeed, neighboring vertices could be removed in the same pass as long as it does not occur at the same time. Since neighboring vertices may be candidates for removal in different threads, it becomes necessary to implement vertex removal in a thread safe manner. This is the approach implemented and Figure 2(b) illustrates our algorithm. This is an alternative to the implementation of a parallel algorithm for creating IS_v. With our approach, it means that multiple threads may perform vertex removals on the same triangulation, while maintaining the integrity of the data structure. If competing threads perform conflicting vertex removals, this situation must be detected and resolved. In our implementation, threads detecting a conflict will ignore the current vertex removal and continue with another candidate vertex.

Serial simplification algorithms can iteratively remove the vertex with lowest decimation cost. The decimation cost can be updated as the neighborhood is modified, ensuring that the priority queue is up to date. In a parallel setting, the use of a priority queue would be a bottleneck, since any update would require exclusive access. Algorithms based on IS_v are guided by the decimation cost, but even the vertex with highest decimation cost can enter IS_v in the first pass. Requiring the candidate vertices to have a decimation cost less than a given threshold can improve this situation. In our algorithm, each thread performs the vertex removals in the order given by the decimation cost. However, the vertices are not reordered within one simplification pass. Depending on the objective of the decimation process, a well suited decimation criterion can prevent unwanted vertex removals, making the application less dependent of the decimation order and on the concept of threshold. Again, this shows what key concepts helped get around the use of IS_v.

Vertices not removed in the first simplification pass may be removed at a later pass. We use a *removable flag* assigned to each vertex, indicating the status of the vertex removal. Vertices failing the simplification criteria are marked as "not removable". This is not a permanent status, since further simplifications can change their neighborhood, hence their cost and status. However, a vertex failing the simplification criteria once is less likely to be removed than vertices not yet considered. As a compromise, we reconsider all vertices marked as "not removable" every third pass. To restrict the SPEs to only consider the removable vertices as candidate vertices, we assign removed and not removable vertices cost of -2 and -1 respectively. After the vertices are sorted in ascending order, the removable vertices are located at the top of the array.

A wide variety of data structures can be used to describe and store triangulations. Due to its simplicity and efficiency, vertex incidence lists are often used for simplifications. In its simplest form, each vertex contains a list of pointers (or indices) to its neighboring vertices. A vertex removal operation performed on this data structure modifies not only the removed vertex V, but also to the vertices in its one-ring neighborhood because the adjacency relations must be updated. Thus, no other vertex in its two-ring neighborhood can be removed at the same time. Thread safe vertex removal operations using such a data structure has an increased risk of failing, especially when the number of vertices gets small, which is often the case when a fair proportion of vertices are removed.

Another popular data structure for triangulations is based on half-edges. Figure 3 illustrates the data elements involved in a vertex removal. The edges surrounding the one-ring neighborhood of a vertex V stay unchanged during its removal. Therefore, we only read and modify the half-edges on the "inside" of this area during a vertex removal. These half-edges are only relevant for vertex removals of any vertex inside this one-ring neighborhood. Therefore, the half-edge data structure does not increase the radius of influence beyond where the triangles change.

Vertex removal operators can result in a locally topologically illegal configuration, such as two faces being mapped on top of each other. Before removing V it must be verified that no edge is added between already connected vertices. To perform this test, the area of interest is expanded from the one-ring neighborhood of V to the two-ring. For the (half-)edge collapse the area of interest is expanded from the one-ring neighborhood of the V to also include the one-ring neighborhood of the destination vertex V_d. In

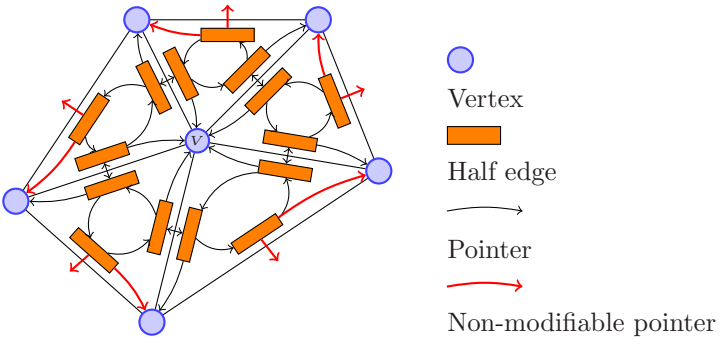


Fig. 3. Illustration of the half-edge data structure. Pointers belonging to boundary edges and therefore not modifiable are marked in red.

Listing 1. Thread safe vertex removal

```
Lock the potential vertex V
Check for conflicts
  Verify that the half-edges ''return'' to the center vertex V
  Verify that none of the neighboring vertices are locked
Check geometrical (and mechanical) criteria
Check topological criteria
Update data structure
  Update pointer to half-edges
  Update the pointers to the vertex V
unlock the potential vertex V
```

Figure 1c) the original area of interest is labeled A_1 and the added area labeled A_T. Due to the small area of interest and simple implementation, half-edge collapse is used in this work.

A thread safe vertex removal operation must detect if either the vertex itself or one of its neighbors is being removed by another thread. Thread safe locking mechanisms are available for most parallel architectures, and we added a lock for each vertex to the data structure to implement such a mechanism. As shown in Listing 1, we lock the potential vertex V as the first step of a vertex removal. During the update of the data structure, we update the pointers to V as the last step, to ensure that competing vertices removal threads see that V is locked. During a vertex removal with the half-edge collapse operator, the data structure is locally inconsistent within the one-ring neighborhood of V. The inconsistencies take form either as three consecutive half-edges not forming a loop, i.e. a triangle is modified or collapses, or half-edges pointing at the incorrect vertex, i.e. the pointer to the half-edges are updated but not the pointer to the vertex. These inconsistencies can easily be detected, by verifying that there is no vertex locked and that three consecutive vertices form a triangle. If neither inconsistency or locked vertices are found, the vertex removal process can safely continue.

4 Cell BE Implementation

From a programming point of view, memory management related issues constitute the main differences between traditional homogeneous architectures and heterogeneous processors such as GPUs and the Cell BE. So far, we have described vertex removal operations using shared memory architectures with coherent caches. Traditional homogeneous multi-processor and multi-core systems fit this description. However, heterogeneous architectures such as GPUs and the Cell BE are not equipped with coherent caches. Therefore, algorithms implemented on such architectures should not rely on memory transfers being performed in the order they are issued. In this section we present our Cell BE implementation of thread safe vertex removal.

Instead of a hardware controlled cache, each SPE has 256KB of local memory called local store, used to store both code and data. A program running on a SPE does not access system memory directly, but uses Direct Memory Access (DMA) instructions to copy data to and from system memory. DMA transfers are asynchronous, allowing the SPE to continue execution while data is being transfered. One way of managing the data transfers is by creating a software managed cache, using DMA transfers to and from local store. Such a cache normally contains functions to read/write data, initiate reading of a memory location (touching), and notifying to the cache that the content may be outdated (dirtying). In addition to DMA instructions, each SPE has an atomic unit. The main purpose of this unit is to allow atomic operations such as locking mechanisms. The atomic unit can also be used to create a software-based coherent cache. However, one should try to avoid the use of the atomic unit, as it easily can become a bottleneck in the application. Instead, we use a software-based cache for reading half-edges and vertices, and perform the corresponding write operations without using the software-based cache. Our modified algorithm uses the atomic unit for maintaining the status of a vertex. A vertex can have the following states: locked, removed, not removable, or removable, covering the needs for both the current decimation pass and the process between the passes.

An implication of lacking coherent cache is that we have no guarantee that memory transfers are performed in the order they are issued. The Cell BE has an option to obey the order of the issued commands, but this postpones each memory operation, causing unnecessary stalls. The basic data element in a SPE is a quadword, and data transfers of naturally aligned quadwords are performed as atomic operations. Based on the concept of atomic operations, we therefore store vertices and half-edges in quadwords to guarantee that we never read a partly updated vertex or half-edge. An half-edge is represented as three indices (four if face attributes are explicitly represented), while a vertex is represented as three floating point numbers and one index. Vertices and half-edges can therefore be stored in a quadword each, with no or little overhead. The updated algorithm is given in Listing 2.

Only pointers pointing towards deleted elements are modified, thus all partly updated triangles will have references to deleted elements (see Figure 3). Therefore, partly updated triangles will either include a reference to a vertex marked for removal, a half-edge marked for removal or the half-edges will not form a loop. If any of these cases are detected, the vertex removal process ignores V, otherwise no conflict is present and the vertex removal continues. Since our cache is not coherent, data may remain in a SPEs

Listing 2. Simplified view of thread safe vertex removal on Cell BE

```
Lock the potential vertex V
While looping around the vertex V to collect neighbors
  Verify that the half-edges ‘‘return’’ to the center vertex V
  Verify that none of the neighboring vertices are locked
Check geometrical (and mechanical) criteria
Check topological criteria
Update data structure
Update the status of V
```

cache after its value is updated. Our data structure consistency tests will detect the cases where the data is partly updated and can either ignore the candidate vertex, or dirty the cache and re-read the inconsistent data.

In our test cases, the overall cache-hit ratio is 80–90%. The high cache-hit ratio is due to the fact that geometrically near data elements are likely to be located near each other in memory. In our test cases each triangle is initially defined by three subsequent half-edges. However, for the first vertex and half-edge read for a given candidate vertex V the cache-hit ratio is very low. This is because the candidate vertices are not treated in their natural order, but rather in a sorted order. Thus, the cache-hit ratio is almost unaffected by dirtying the cache at the beginning of each vertex removal operation.

The cost of a cache-miss is reduced if the processor is busy while the memory transfer takes place. This can be achieved by touching the cache before the data is needed. This strategy is only feasible if the program can continue execution before the data is transferred. In our algorithm this is not the case. Bader et al. [16] presented an alternative latency-hiding technique for the Cell BE. They use software-managed threads to let the SPE continue working after a memory request is issued. The SPEs do not have hardware support for quickly switching between threads. Therefore, the program itself is responsible for switching between the software-managed threads. We applied this strategy in our application, manually switching threads after a cache miss. To facilitate this behavior, we implemented two different cache touching functions in our software-based cache. One function only starts the required data transfer, while the other function also reports if the data element is already in cache. This allows us to only swap threads when an actual cache-miss is encountered. However, with the high cache-hit ratio in our algorithm this only gave us 2% speedup.

5 Results

The main purpose here, is to highlight how the speedup scales up when the number of threads is increasing. For each parallelizable stage of our algorithm, we uniformly divide the vertices among the threads. The computation time seems to be sufficiently uniform to justify this choice. We compared our cell implementation with an unoptimized version ran on the PPE only. In our benchmarks the PPE version and the SPE version ran neck and neck. This indicates that our pre-fetching strategy and software based cache is successful. However, no strong conclusions should be made based on this, since the PPE version may be far from optimal.

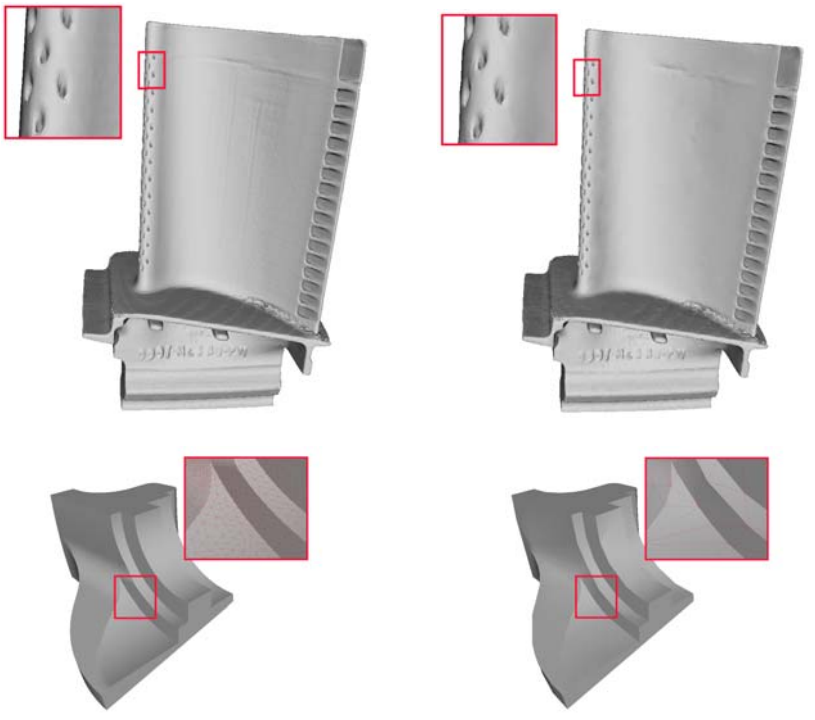


Fig. 4. Turbine blade (top) and fandisk model (bottom) before and after simplification

Figure 4 shows the two test models before and after simplification. The vertex-cost is set to its discrete Gaussian curvature. The 90% vertices with lowest costs are considered as potential for removal. Potential vertices are removed if the maximal angular deviation from the previous model is less than 25 degrees and all interior angles in the resulting triangles are larger than 10 degrees. This yields good visual quality of the resulting triangulations. However, they do not provide any guaranteed error estimates. These criteria were chosen because of their balance between computations and memory operations. In addition, we include benchmark results where only topology tests are performed. Obviously, the resulting models are not usable, since the geometry is ignored. However, it shows that our algorithm performs well also for less computationally intensive decimation criteria.

In the current Cell BE implementation the sorting is performed by the PPE. Parallel sorting algorithms are becoming a standard component for parallel architectures, and is a part of software development kits for GPUs and the standard C++ library distributed with the `gcc` compiler now contains parallel sorting. However, sorting is not yet a part of the IBM Cell SDK [17] we used for the Cell BE. Implementing parallel sorting algorithms such as AA-sort by Inoue et al. [15] is outside the scope of our work. We therefore performed the benchmarks using a single-threaded sorting algorithm. For the turbine blade model in Figure 4, sorting and other sequential parts of the code represent

less than one percent of the total runtime when using one SPE. The lack of parallel sorting implementation does not notably affect this benchmark, but must be remedied if the number of cores is to increase greatly.

Figure 5 illustrates the speedups achieved using a varying number of SPEs. The corresponding runtimes using one SPE is listed in Table 1. For the blade model, the speedup is almost linear for both benchmarks. The much smaller fandisk model does not achieve the same speedup when only topology is tested. This is due to the overhead caused by starting and stopping SPE threads.

Table 1. Runtimes of simplification, using one SPE on IBM QS21

model	#vertices	runtime geometric	runtime topology
blade	882954	53.7s	9.0s
fandisk	6475	0.37s	0.05s

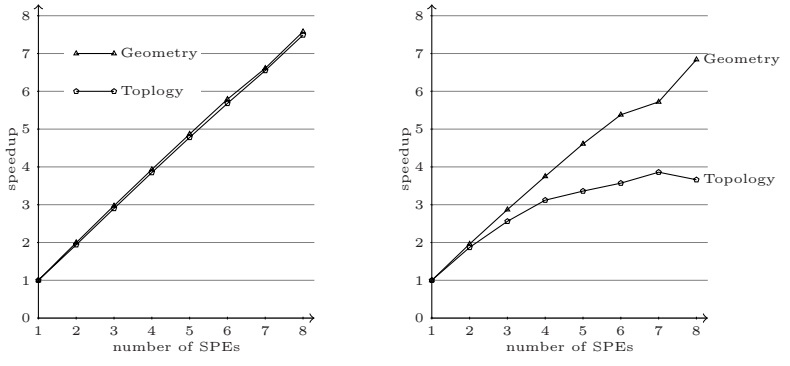


Fig. 5. Speedup of simplification of the turbine blade model (left) and the fandisk model (right). Benchmark of geometry-based simplification criteria and purely topological ones are presented.

6 Concluding Remarks

We have developed a flexible framework for shape simplification, suitable for heterogeneous many-core systems. The vertex removal is performed in a thread safe manner, allowing each thread to operate independently and eliminating the need of an independent set of vertices. The flexibility of the framework allows the inclusion of any decimation criterion based on information located inside the one-ring neighborhood. Furthermore, properties associated to the faces can be propagated throughout the simplification process, thus allowing simplification criteria with guaranteed error estimates such as error spheres used by Véron and Léon [18].

Our benchmarks have shown that our framework performs well both for arithmetically intensive and memory intensive decimation criteria. Furthermore, it has sufficiently low overhead to be used in applications treating a large number of smaller models. The

use of software-managed threads did not give us the expected performance increase. This may change if the software related to software-managed threads is optimized.

The strategy presented here can be used also for other operations performed on triangulations. Operations such as edge-flip and the Bowyer Watson algorithm for vertex insertion are candidates for this framework.

GPUs supporting the CUDA API perform data transfers of naturally aligned quadwords as atomic operations. Locking mechanisms are now also available on commodity GPUs. This opens up the possibility to develop a GPU-based implementation of our framework. Whether or not such an implementation is feasible is an open question, as the execution units of current GPUs operate in a synchronous manner. We will investigate this in a future article.

References

1. Batcher, K.E.: Sorting networks and their applications. In: AFIPS Spring Joint Computing Conference, pp. 307–314 (1968)
2. Purcell, T.J., Donner, C., Cammarano, M., Jensen, H.W., Hanrahan, P.: Photon mapping on programmable graphics hardware. In: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware, pp. 41–50. Eurographics Association (2003)
3. Owens, J., Houston, M., Luebke, D., Green, S., Stone, J., Phillips, J.: GPU computing. Proceedings of the IEEE 96(5), 879–899 (2008)
4. Gotsman, C., Gumhold, S., Kobbelt, L.: Simplification and compression of 3d meshes. In: Tutorials on Multiresolution in Geometric Modelling, pp. 319–361. Springer, Heidelberg (2002)
5. Schroeder, W.J., Zarge, J.A., Lorensen, W.E.: Decimation of triangle meshes. In: SIGGRAPH 18992: Proceedings of the 19th annual conference on Computer graphics and interactive techniques, pp. 65–70. ACM, New York (1992)
6. Hoppe, H.: View-dependent refinement of progressive meshes. In: SIGGRAPH 1997: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp. 189–198. ACM Press/Addison-Wesley Publishing Co., New York (1997)
7. Kobbelt, L., Campagna, S., peter Seidel, H.: A general framework for mesh decimation. In: Proceedings of Graphics Interface, pp. 43–50 (1998)
8. Dadoun, N., Kirkpatrick, D.G.: Parallel algorithms for fractional and maximal independent sets in planar graphs. Discrete Appl. Math. 27(1-2), 69–83 (1990)
9. Karp, R.M., Wigderson, A.: A fast parallel algorithm for the maximal independent set problem. In: STOC 1984: Proceedings of the sixteenth annual ACM symposium on Theory of computing, pp. 266–272. ACM, New York (1984)
10. Franc, M., Skala, V.: Parallel triangular mesh decimation without sorting. In: SCCG 2001: Proceedings of the 17th Spring conference on Computer graphics, Washington, DC, USA, p. 22. IEEE Computer Society, Los Alamitos (2001)
11. Botsch, M., Bommes, D., Vogel, C., Kobbelt, L.: GPU-based tolerance volumes for mesh processing. In: PG 2004: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference on (PG 2004), Washington, DC, USA, pp. 237–243. IEEE Computer Society, Los Alamitos (2004)
12. Hjelmervik, J., Léon, J.C.: GPU-accelerated shape simplification for mechanical-based applications. In: Shape Modeling International, pp. 91–102 (2007)
13. DeCoro, C., Tatarchuk, N.: Real-time mesh simplification using the GPU. In: I3D 2007: Proceedings of the 2007 symposium on Interactive 3D graphics and games, pp. 161–166. ACM, New York (2007)

14. Govindaraju, N., Gray, J., Kumar, R., Manocha, D.: GPUTeraSort: high performance graphics co-processor sorting for large database management. In: SIGMOD 2006: Proceedings of the 2006 ACM SIGMOD international conference on Management of data, pp. 325–336. ACM, New York (2006)
15. Inoue, H., Moriyama, T., Komatsu, H., Nakatani, T.: AA-sort: A new parallel sorting algorithm for multi-core SIMD processors. In: PACT 2007: Proceedings of the 16th International Conference on Parallel Architecture and Compilation Techniques, pp. 189–198. IEEE Computer Society, Los Alamitos (2007)
16. Bader, D.A., Agarwal, V., Madduri, K.: On the design and analysis of irregular algorithms on the Cell processor: A case study of list ranking. In: Proc. of the 21st International Parallel and Distributed Processing Symposium, pp. 1–10. IEEE, Los Alamitos (2007)
17. IBM: Software development kit for multicore acceleration version 3.1: Programmers guide (August 2008)
18. Véron, P., Léon, J.C.: Shape preserving polyhedral simplification with bounded error. Computers & Graphics 22(5), 565–585 (1998)

Effects of Noise on Quantized Triangle Meshes

Ioannis Ivrissimtzis

Durham University, Durham, UK
ioannis.ivrissimtzis@durham.ac.uk
http://www.dur.ac.uk/ioannis.ivrissimtzis

Abstract. In this paper we measure the effects of noise on the performance of a standard entropy compression algorithm for quantized triangle meshes. In a first experiment, we show that a small amount of added noise may lead to slightly smaller filesizes. The magnitude of stochastic resonance is too small to have any practical applications in mesh compression. In a second experiment, we study the effect of a fixed amount of added noise on different levels of quantization. The results show that the effect of the added noise does not increase monotonically with the level of quantization. These two phenomena have implications in the design of validation experiments for mesh processing algorithms. They show that the tacit assumption that the effects of added noise increase monotonically with the amount of noise and with the level of quantization can be wrong.

1 Introduction

Three dimensional models reconstructed from scan data are now commonly used in graphical applications. Nevertheless, and despite the improvements in the optical data acquisition technology, imperfections in the whole process mean that noise is always present. Noise affects negatively the quality of a model in a way which can be described, depending on the point of view, as information loss, or pattern degradation, or entropy increase. By common practice, which is of course theoretically justified, we generally assume that the negative effects on a model's quality increase monotonically with the volume of the noise.

In this paper, we argue that this assumption is a simplification which may not hold within the context of entropy compression of quantized triangle meshes. To study the response of the compression rate to added noise, we first encode the mesh connectivity and then quantize the vertex coordinates and encode them with a predictive scheme. The obtained corrective vectors are entropy encoded. Comparing the filesizes, we often detect a small reduction when small amounts of noise are added, see Fig. 1.

We explain this behavior through the *stochastic resonance* paradigm. Stochastic resonance describes the phenomenon where small amounts of noise improve the ability of biological or electronic systems to detect a signal [1]. Typically, it appears when a system can only detect signals that are above a threshold. In this case, a small amount of added noise may slightly enhance the signal and push it just above the system's detection threshold. Stochastic resonance has been studied theoretically and has been detected in various biological systems and computer simulations.

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 274–284, 2010.

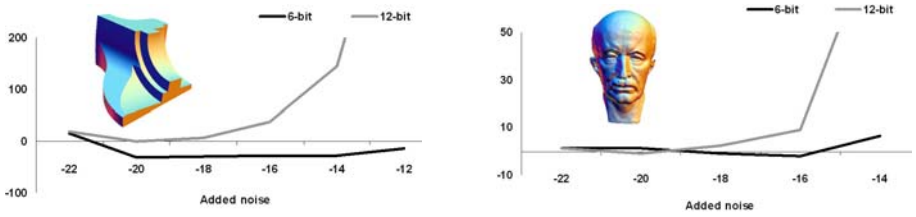


Fig. 1. The *x*-axis shows the amount of added noise as a power of 2. The *y*-axis shows the average difference in the filesize in bytes. Small amounts of noise reduce the filesize.

In entropy mesh compression, all the main components of the stochastic resonance paradigm are present. Indeed, the positions of the mesh vertices are the signal. The most significant bits represent the low frequencies and the least significant bits the high frequencies and we only encode the low frequencies with the quantization procedure determining a detection threshold. Finally, the use of entropy encoding means that the resulting filesize is a measure of the signal's entropy.

In a second experiment, we study the effect of a fixed amount of added noise at different levels of quantization. If the initial model was noise free, one would expect the larger effect of the added noise to be on the model's high frequencies, that is, at the finest levels of quantization. However, because the initial model already contains noise, as the frequencies increase, their entropy tends to the theoretical maximum and the effect on them of the additional noise tends to zero. As a result, starting from the low frequencies, the effect of a fixed amount of added noise is initially small, then as the frequencies increase the effect of the added noise increases and reaches a maximum, and finally, at the highest, already corrupted frequencies, the effect of the added noise decreases again. This behavior was verified by the second experiment.

Summarizing, the first contribution of the paper is to demonstrate that within the context of entropy mesh compression, a small amount of noise can reduce the filesize of the compressed model. The second contribution is to propose an interpretation of this phenomenon as a manifestation of stochastic resonance in quantized triangle meshes. The third contribution is to demonstrate that, depending on the initial noise of the model, some intermediate levels of quantization are the most sensitive to a fixed amount of added noise.

1.1 Previous Work

The literature on stochastic resonance is extensive, reflecting the wide range of areas where such phenomena have been detected. [2] is one of the early papers studying stochastic resonance. [3] gives a detailed theoretical study of the phenomenon. An extended review of the subject can be found in [4].

Surfaces and triangle meshes in particular have been studied from a signal theoretic point of view for a long time. Two influential early papers are [5] with a traditional signal-theoretic approach, and [6] with a wavelet-subdivision approach. Applications of the signal theoretic approach to triangle mesh compression have been studied in [7, 8].

Noise and uncertainty in 3d models has also attracted considerable research interest and a detailed study of the topic, in the context of point clouds, can be found in [9].

2 The Implementation of the Experiments

Following a typical mesh compression process we first encode the connectivity of the model. Then, the vertex coordinates are quantized and encoded with a predictive scheme and finally, the corrective vectors are entropy encoded.

For connectivity encoding we use the Edgebreaker [10]. It is a widely used algorithm for triangle mesh connectivity compression giving compression rates comparable with the theoretical bound of 3.24 bits per vertex. As spatial noise does not affect the connectivity of a mesh, we are not directly interested in the entropy of the connectivity encoding and we do not report any results on the filesize of the compressed connectivity. However, the connectivity encoding implies a traversal of the mesh triangles, which traversal affects the encoding of the vertex coordinates.

For the predictive encoding of the vector of the vertices $\bar{\mathbf{v}}$ we use the parallelogram rule [11]. Because of its simplicity, this rule is still considered the state of the art, even though many other more complicated predictive algorithms have shown improved results. Finally, the corrective vector is encoded with a version of arithmetic encoding [12]. Arithmetic encoding generally gives better compression ratios than the simpler alternative of Huffman encoding. In the experiments, we use implementations of the Edgebreaker and the arithmetic coder that are available online [13,14].

Before compressing the model, we add uniform random noise to $\bar{\mathbf{v}}$. For a better insight into the experimental results, especially those in Section 4, we assume a probabilistic model for the vertices of the mesh. The probability distributions of the vertices are considered independent variables. The mean of the probability distribution of the vector of vertices $P(\bar{\mathbf{v}})$ is equal to $\bar{\mathbf{v}}$. The addition of uniform noise gives a new distribution

$$P(\bar{\mathbf{v}}) \otimes U^3[-d/2, d/2] \tag{1}$$

with the same mean, which is the convolution of the initial distribution with the 3d uniform distribution with mean (0,0,0) and support d^3. Notice that the entropy of the distribution in Eq. 1 is always higher than the entropy of $P(\bar{\mathbf{v}})$. We will refer to d as the amount of added noise.

After adding uniform noise, the model is rescaled to have the unit cube as its bounding box, and quantized. Notice that in almost all graphical applications the vertex coordinates, either represented by floats or by integers, are discrete. That is, they are usually described by a certain number of bits while, in contrast, the use of exact arithmetic is rare. That means that our quantization of the model is not a discretization, but instead it is a thresholding applied on an already discrete representation.

In the experiments we used four well-known models: the *Horse*, *Fandisk*, *MaxPlanck* and *Venus*. For each original model we created two more versions of it. The noisy version was created by adding uniform noise with $d = 5 \times 10^{-4}$, assuming that the bounding box of the original is the unit cube. The smooth version was created by 50 iterations of the umbrella operator on the original model with parameter $\lambda = 0.5$. The models are shown in Fig. 2.

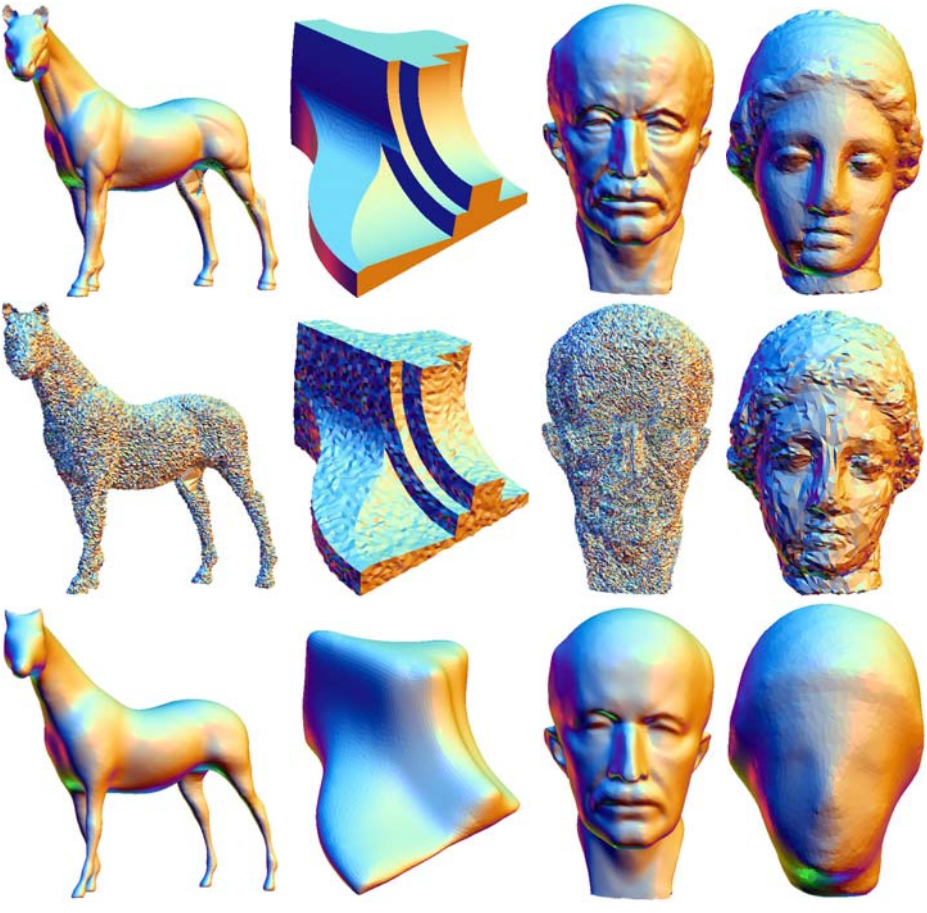


Fig. 2. The twelve models used in the experiments

3 Stochastic Resonance in Quantized Triangle Meshes

The results of the first experiment are are shown in Tables 1-4 in the Appendix. For each model, level of quantization and amount of added noise, we did 20 experiments. In each case we give the mean and the standard deviation of the differences from the filesize of the original, reporting in bytes. In most cases, we notice a small reduction in the filesize when small amounts of noise are added.

The reduction of the filesize can be seen as the manifestation of stochastic resonance. In a broader interpretation, the non-monotonicity of the filesize as a function of the added noise, can also be seen as a manifestation of stochastic resonance. For example, in the 14-bit smooth Fandisk model shown Fig. 3 (left) the added noise always increases the filesize of the original. Nevertheless, we can detect stochastic resonance

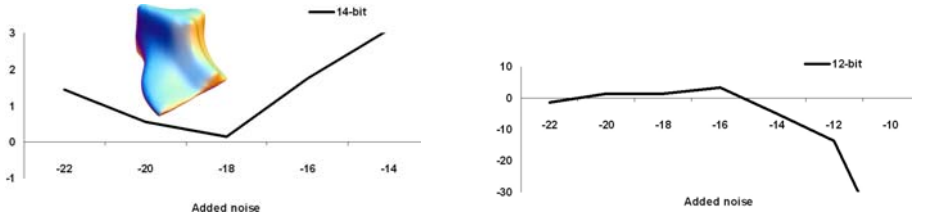


Fig. 3. Left: Compressed filesize of the 14-bit smooth Fandisk. **Right:** When the vertex coordinates are totally random, the curve of the filesize is inverted.

in the broader sense as added noise $d = 2^{-18}$ leads to a smaller filesize compared to $d = 2^{-20}$, or $d = 2^{-22}$.

To justify this broader interpretation of stochastic resonance, we notice that, usually, the original model already contains noise. Thus, the filesize reported for $d = 0$ is just an estimate of the mean filesize for the initial amount of noise, and thus, it should not be treated as the filesize of a noise free model. In other words, the effects of the added noise are better described by the function of the filesize in terms of the added noise, in particular its local minima, than by a comparison with the filesize of zero added noise.

In a final experiment, we substituted all the vertices of the mesh with vertices with random uniform distribution $U^3[-1/2, 1/2]$. The results show an inversion of the typical filesize curve, see in Table 5 in the Appendix, and Fig. 3 (right). This last experiment is interesting because we know the exact distribution $P(\bar{\mathbf{v}})$ and can compute its entropy. Indeed, with $d = 0$, each vertex follows the $U^3[-1/2, 1/2]$ distribution and thus has maximum entropy. By adding a certain amount of noise $U^3[-d/2, d/2]$ the distribution of each vertex becomes the convolution of two uniform distributions $U^3[-1/2, 1/2] \otimes U^3[-d/2, d/2]$, rescaled so that its support is the interval $[-1/2, 1/2]^3$. As the uniform distribution has maximum entropy, the new distribution has lower entropy. We can easily check that the minimum entropy of the convolution of two uniform distributions, rescaled to the interval $[-1/2, 1/2]^3$, is obtained when $d = 1$, in which case we get a triangle distribution. Then, as d increases above 1, the added noise starts to dominate the original signal and the entropy increases again.

From these theoretical entropies, one would expect that, as d increases, the filesize would decrease, until $d = 1$, and then it would start increasing again. Instead, we notice that for a small d, even though the theoretical entropy of the model decreases, the filesize increases. We can see this as a manifestation of stochastic resonance, in the sense that the thresholding induced by the quantization changes the expected behavior of the entropy. We consider this last experiment to be the most conclusive detection of stochastic resonance in quantized triangle meshes, as it is the only case where the distribution $P(\bar{\mathbf{v}})$ is known.

Even though unrelated to the phenomenon of stochastic resonance, in a final observation from Tables 1-4, we notice that the filesize of the Fandisk and the Horse models increased with smoothing. In the case of Fandisk, smoothing degraded patterns of flat surfaces meeting at sharp edges. In the case of Horse, smoothing degraded patterns caused by the regular sampling of the object's surface by the acquisition device. In contrast, the vertices of the Max-Planck and the Venus models are processed subsamples of

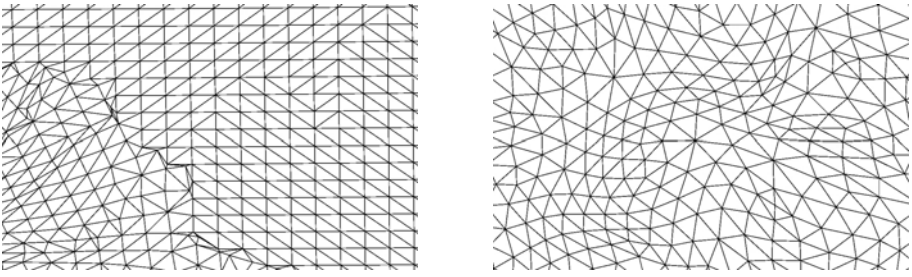


Fig. 4. Wireframe views of parts of the Horse (left) and the Max-Planck (right) models

the originally acquired geometry and they do not exhibit this kind of regular patterns, see Fig. 4.

4 Effect of Added Noise at Different Levels of Quantization

The magnitude of the phenomenon of stochastic resonance, as measured by the first experiment, is negligible. We typically observe differences of few bytes on total filesizes of several kilobytes. This can be explained by the large size and the irregularity of the triangle meshes used in the experiments. Patterns just below the quantization threshold are pushed above it by the added noise, but this effect is counterbalanced by patterns just above the quantization threshold pushed below it. In contrast, the second experiment studies a phenomenon of larger scale, which again is related to changes in entropy by added noise. In particular, it studies how the added noise affects the entropy at different levels of quantization.

In this case, as we compare different levels of quantization, and as the level of quantization also affects the filesize, we do not report the filesize differences in bytes, but as percentages of the initial filesize. That is, if s, s' are the filesizes for a given quantization level before and after the addition of noise, we report the relative difference $(s' - s)/s$. Fig. 5 shows these relative differences for different levels of quantization and for different amounts of added noise. We notice that, in some cases, the relative difference first increases with the level of quantization and then decreases, giving a local maximum.

This local maximum is related to the noise of the initial model. Indeed, if the initial model was noise free we could assume that the appearance of compressible patterns is independent of the level of quantization and thus, the rate of compression r is independent of the level of quantization. The effect of the added noise on the entropy of the noise free mesh would increase monotonically with the level of quantization, the compression rate r' of the model with added noise would monotonically decrease and tend to 1, while the term $r/r' - 1$ would monotonically increase and tend to $r - 1$. For a given level of quantization, the uncompressed meshes before and after the addition of noise have the same filesize, thus, the relatively difference $(s' - s)/s = s'/s - 1 = r/r' - 1$ would also monotonically increase and tend to $r - 1$.

In contrast, if the original model does have some noise, as the level of quantization increases the entropy tends to its maximum and thus the effect of the added noise is

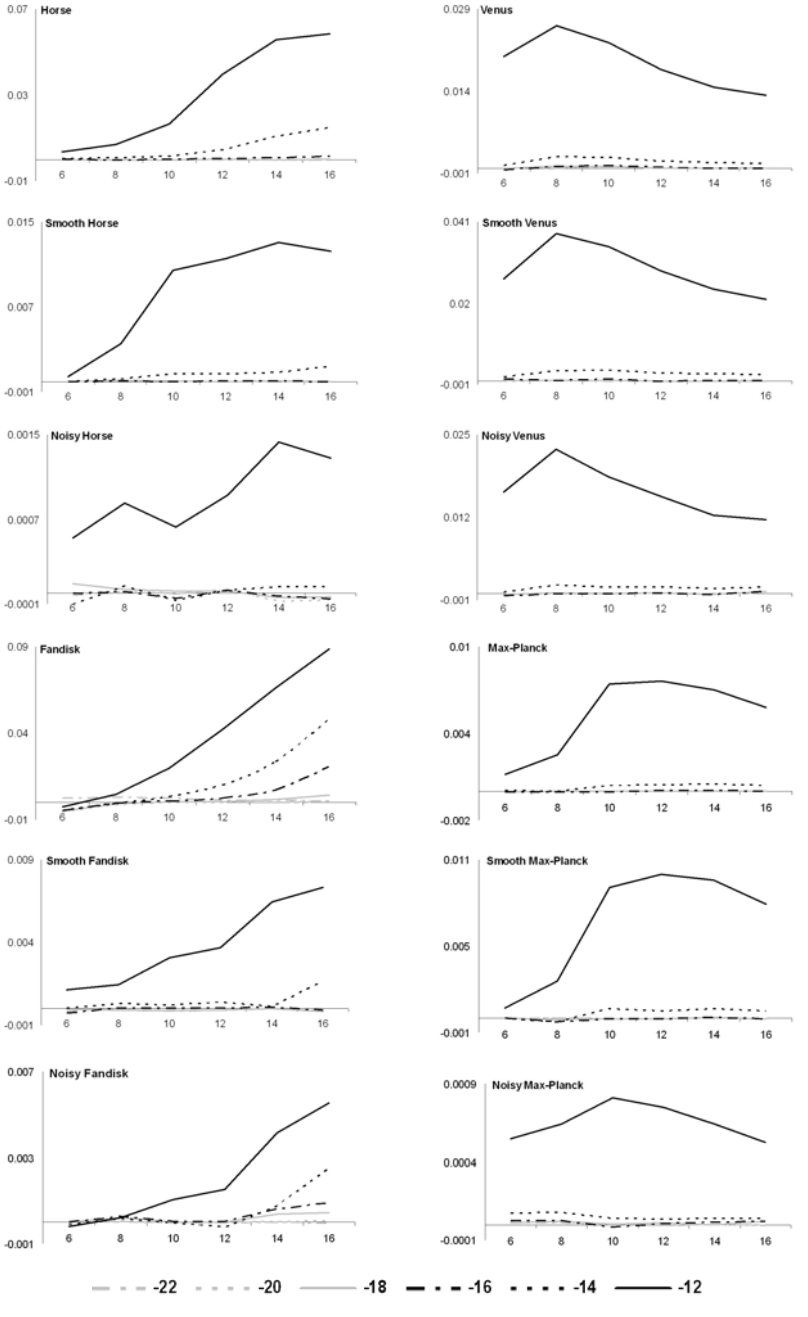


Fig. 5. Each line of a graph corresponds to a different amount of added noise, from $d = 2^{-22}$ to $d = 2^{-12}$. The x-axis corresponds to the level of quantization and the y-axis to the relative difference of the filesizes.

minimal. That means that the ratio r/r' tends to 1, and $(s' - s)/s = r/r' - 1$ tends to zero. As a result, the largest effect of the added noise is not on the highest levels of quantization but at some intermediate levels, which they are not very coarse, and thus their entropy is affected considerably by the added noise, but not very fine either, and thus their entropy is not already near its maximum.

In Fig. 5 we see such local minima, especially for the Venus and Max-Planck models. That was expected as the Fandisk and the Horse models have low entropy, that is, low initial noise, while the local minimum is a manifestation of the initial noise. In the latter two cases, even though we can not detect a local minimum, still we can see that the smooth and the noisy versions, which both have higher entropy than the original, tend nearer towards that behavior.

Notice that for visualization purposes the scale of the y-axes in Fig. 5 varies from graph to graph. For example, the noisy versions of the models always have lower values of relative filesize differences than the originals, even when the shapes of the corresponding curves are similar.

5 Discussion

We measured the effects of added noise on the filesize of entropy compressed quantized triangle meshes. In a first experiment, we showed that usually the filesize does not increase monotonically with the amount of added noise. Instead, when small amounts of noise are added, we may have a small reduction in the compressed filesize. We argue that this phenomenon should be interpreted inside the wider paradigm of stochastic resonance. In a second experiment, we showed that the effects of the added noise on the filesize do not always increase monotonically with the level of quantization. Indeed, the effect of the added noise at the finest levels of quantization could be minimal. We argue that this phenomenon is related to the initial noise of the model.

In the future we plan to investigate the effect of stochastic resonance on shape descriptors, especially those based on coarse quantizations, as for example the ones described in [15] which use 6-bits per coordinate. More generally, the measurement of the effects of different amounts of added noise on different frequencies of a quantized mesh can be seen as a shape analysis algorithm itself. We plan to investigate whether we can analyze the shape of a mesh through the responses of its different frequencies to different levels of added noise. Such experiments would mimic the way we study physical models through their resonance to physical stimuli of different frequencies and amplitudes.

References

1. Wiesenfeld, K.: Stochastic resonance and the benefits of noise: from ice ages to crayfish and SQUIDs. Nature 373, 33–36 (1995)
2. Benzi, R., Sutera, A., Vulpiani, A.: The mechanism of stochastic resonance. J. Phys. A 14, L453–L457 (1981)
3. Greenwood, P.E., Müller, U.U., Ward, L.M., Wefelmeyer, W.: Statistical Analysis of Stochastic Resonance. Austrian Journal of Statistics 32, 49–70 (2003)

4. Gammaitoni, L., Hänggi, P., Jung, P., Marchesoni, F.: Stochastic resonance. Rev. Modern Phys. 70, 223–288 (1998)
5. Taubin, G.: A signal processing approach to fair surface design. In: SIGGRAPH 1995, pp. 351–358. ACM Press, New York (1995)
6. Schröder, P., Sweldens, W.: Spherical wavelets: efficiently representing functions on the sphere. In: SIGGRAPH 1995, pp. 161–172. ACM Press, New York (1995)
7. Karni, Z., Gotsman, C.: Spectral compression of mesh geometry. In: SIGGRAPH 2000, pp. 279–286. ACM Press, New York (2000)
8. Sorkine, O., Cohen-Or, D., Toledo, S.: High-pass quantization for mesh encoding. In: Proceedings of the Symposium on Geometry processing, pp. 42–51. Eurographics Association (2003)
9. Pauly, M., Mitra, N.J., Guibas, L.: Uncertainty and variability in point cloud surface data. In: Symposium on Point-Based Graphics, pp. 77–84 (2004)
10. Rossignac, J.: Edgebreaker: Connectivity compression for triangle meshes. IEEE Transactions on Visualization and Computer Graphics 5(1), 47–61 (1999)
11. Touma, C., Gotsman, C.: Triangle mesh compression. In: Proceedings of the 24th Conference on Graphics Interface, pp. 26–34 (1998)
12. Moffat, A., Neal, R., Witten, I.: Arithmetic coding revisited. ACMTOIS: ACM Transactions on (Office) Information Systems 16, 256–294 (1998)
13. Safonova, A.: http://www.gvu.gatech.edu/~jarek/edgebreaker/eb/
14. Moffat, A.: http://www.cs.mu.oz.au/~alistair/arith_coder/
15. Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S., Dobkin, D.: Modeling by example. ACM Trans. Graph. 23(3), 652–663 (2004)

Appendix

Table 1. Fandisk: The filesizes reported in bytes

Added noise d		0	2^{-22}	2^{20}	2^{-18}	2^{-16}	2^{-14}	2^{-12}
6-bit	Mean	6148	16	-31.7	-30.1	-27.2	-27.6	-14.3
	Stdv	0	5.5	3.8	2.7	3.5	5.1	7.0
12-bit	Mean	14169	19	-0.45	6.8	37.15	145.2	600
	Stdv	0	3.6	1.4	2.2	5.7	7.7	9.8
Smooth 6-bit	Mean	6129	-1.1	-0.15	-0.3	-1.55	0.45	7.15
	Stdv	0	1.0	1.35	1.3	2.0	3.8	7.5
Smooth 12-bit	Mean	15572	-1.4	-1.4	-1.55	0.75	6.55	57.45
	Stdv	0	1.2	1.2	1.6	2.6	3.6	5.0
Noisy 6-bit	Mean	6790	0.5	0.15	0.05	0.2	-0.6	-1.4
	Stdv	0	0.8	1.1	0.9	1.6	4.1	7.3
Noisy 12-bit	Mean	18178	-0.1	0.15	0.3	0.55	-3.85	27.6
	Stdv	0	1.0	1.7	2.5	3.1	4.2	5.6

Table 2. Horse: The filesizes reported in bytes

Added noise d		0	2^{-22}	2^{20}	2^{-18}	2^{-16}	2^{-14}	2^{-12}
6-bit	Mean	41326	0.95	0.2	0.35	2.65	21.85	156.05
	Stdv	0	1.8	1.7	3.7	5.4	10.4	22.7
12-bit	Mean	95396	-1.65	0.35	6.9	49.45	432.75	3791.25
	Stdv	0	2.8	3.5	8.2	13.8	20.8	31.5
Smooth 6-bit	Mean	41640	-0.25	-0.85	-1.65	-0.4	-2.3	17.75
	Stdv	0	1.7	2.4	2.9	4.9	11.0	13.8
Smooth 12-bit	Mean	95429	1.6	-0.5	-0.75	2.0	68.0	1103
	Stdv	0	2.1	2.5	4.2	8.3	11.0	22.2
Noisy 6-bit	Mean	47061	-0.95	0.4	4.25	0.25	-4.55	24.6
	Stdv	0	1.5	2.1	3.6	7.9	17.4	23.1
Noisy 12-bit	Mean	131033	4.65	4.65	2.85	3.65	3.95	121.7
	Stdv	0	3.3	5.0	4.6	4.0	8.2	16.8

Table 3. Venus: The filesizes reported in bytes

Added noise d		0	2^{-22}	2^{20}	2^{-18}	2^{-16}	2^{-14}	2^{-12}
6-bit	Mean	9697	-1.0	-0.55	-0.8	-2.75	5.0	196.75
	Stdv	0	1.1	1.5	2.8	5.3	9.8	17.1
12-bit	Mean	26043	1.45	-0.15	-0.85	3.7	32.5	468.55
	Stdv	0	2.1	2.9	4.3	5.5	7.7	16.2
Smooth 6-bit	Mean	9188	0.55	-0.05	0.6	5.05	9.35	242.25
	Stdv	0	0.9	1.3	2.7	4.5	8.0	18.1
Smooth 12-bit	Mean	24915	0	-0.9	-1.8	0.85	48.85	709.2
	Stdv	0	2.7	3.3	3.3	4.9	6.7	9.6
Noisy 6-bit	Mean	9840	-0.7	-0.55	-1.25	-3.45	2.1	156.85
	Stdv	0	0.9	1.8	3.7	8.2	13.6	14.0
Noisy 12-bit	Mean	26349	1.5	4.05	4.7	4.6	26.45	404.15
	Stdv	0	2.7	3.3	3.6	5.8	5.2	16.3

Table 4. Max-Planck: The filesizes reported in bytes

Added noise d		0	2^{-22}	2^{20}	2^{-18}	2^{-16}	2^{-14}	2^{-12}
6-bit	Mean	84694	1.4	1.2	-0.8	-1.85	6.6	99.8
	Stdv	0	1.4	2.2	6.7	7.4	17.8	36.6
12-bit	Mean	202939	1.35	-1	2.5	9.1	95.95	1552
	Stdv	0	3.7	4.4	9.1	12.0	22.7	26.3
Smooth 6-bit	Mean	83754	1.9	1	0.8	-0.75	1.2	60.85
	Stdv	0	2.0	2.8	3.3	7.2	16.9	34.7
Smooth 12-bit	Mean	196469	1.85	0.75	-6.75	-8.95	98.15	1967.8
	Stdv	0	3.4	5.1	7.4	12.7	20.6	20.6
Noisy 6-bit	Mean	95721	1.3	1.5	1.3	2.55	6.95	52.6
	Stdv	0	1.6	2.0	4.7	7.8	16.2	42.0
Noisy 12-bit	Mean	265090	-1.9	-2	2.8	2.65	9.25	198.6
	Stdv	0	4.6	5.2	6.4	8.4	14.7	20.4

Table 5. Average compressed filesizes of a 12-bit horse with uniformly random coordinates

Added noise d		0	2^{-22}	2^{20}	2^{-18}	2^{-16}	2^{-14}	2^{-12}	2^{-10}	2^{-8}
Random	Mean	261796	-1.25	1.35	1.3	3.45	-4.9	-13.6	-51.55	-207.55
	Stdv	0	3.6	4.8	4.4	7.4	11.6	10.6	13.5	11.8

Reparameterization of Curves and Surfaces with Respect to Their Convolution

Miroslav Lávička, Bohumír Bastl, and Zbyněk Šír

University of West Bohemia, Faculty of Applied Sciences, Department of Mathematics,
Univerzitní 8, 301 00 Plzeň, Czech Republic
lavicka@kma.zcu.cz

Abstract. Given two parametric planar curves or surfaces we find their new parameterizations (which we call coherent) permitting to compute their convolution by simply adding the points with the same parameter values. Several approaches based on rational reparameterization of one or both input objects or direct computation of new parameterizations are shown. Using the Gröbner basis theory we decide the simplest possible way for obtaining coherent parametrizations. We also show that coherent parameterizations exist whenever the convolution hypersurface is rational.

1 Introduction

Computer Aided Geometric Design, and consequently a vast variety of geometrical applications, is based on parametric piecewise rational representations, such as B-splines and NURBS. Therefore, the major problem of CAGD is that many natural geometrical operations do not preserve rationality of derived objects. Among the most frequent such operations belong offsetting of curves and surfaces, convolutions and Minkowski sums. The practical implementation of these operations usually requires approximation techniques which typically lead to a dramatical increase of the number of used elements (pieces in a piecewise representation).

As an attempt to solve those problems, a lively research interest in objects and representations closed under mentioned operations appeared. For example, the concept of planar Pythagorean Hodograph (PH) curves as polynomial curves with rational offsets and polynomial arc-length function was introduced in [4] and later analyzed in many publications, see e.g. [6,14,26]. A spatial variant of PH curves, which possess rational frames, was introduced in [5] and analyzed e.g. in [8,10,28]. For a survey of planar and spatial PH curves see [7].

Another approach is based on dual representations of curves and surfaces. A full characterization of rational hypersurfaces with rational offsets was given in [19,21,22] via description of hypersurfaces as envelopes of tangent hyperplanes. Further, the concept of Minkowski Pythagorean Hodograph (MPH) curves (cf. [15,18]) allows to describe domain boundary as a (rational) envelope of a family of hyperspheres (which form the Medial Axis Transform of the domain). Finally, motivated by the convex geometry, the support function representation of curves and surfaces was used in [1,11,29], where hypersurfaces are described by the distance of their tangent planes taken as a function of the corresponding unit normals.

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 285–298, 2010.

The convolution of hypersurfaces can be seen as a general concept embodying several of above constructions. Indeed, the construction of classical offsets can be seen as a convolution with a sphere, construction of general offsets as a convolution with an arbitrary other surface, etc. Many interesting problems related to this topic have arisen, e.g. analysis of (geometric and algebraic) properties, determining number and kind of their components (cf. [2]), description of their structure (cf. [9]), computing the convolution degrees of hypersurfaces (cf. [16]) and mainly a construction of rational parameterizations of convolution hypersurfaces, if they exist.

This paper is devoted to the analysis of reparameterizations of given hypersurfaces with respect to their convolution. This problem already attracted certain attention, see [13, 16, 20, 23]. Our main result is Theorem 11 which gives a necessary and sufficient condition for the existence of coherent parameterizations and provides a way how to find them. The main idea lies in the description of hypersurfaces by dual (implicit support function) representation and the following symbolic manipulations based on Gröbner bases computations (see [3]).

The remainder of the paper is organized as follows. In Section 2 we give, after some motivation, the definition of coherent parameterizations. In Section 3 we formulate the general strategy of finding a parameterization coherent with given normal field and show how it can be used for solving the problem by changing the parameterization of only one of the given hypersurfaces. We also decide when this new parameterization is rationally related to the original one. In Section 4 we show how to proceed when both parameterizations need to be changed. Again, we distinguish the case when the coherent parameterizations can be obtained via rational reparameterizations of the original ones. In Section 5 we summarize the whole algorithm and conclude the paper.

2 Statement of the Problem

Let A, B be two hypersurfaces in $\mathbb{R}^{n+1}$ where at each point $\mathbf{x}$ of A or B we have a given nonzero normal $\mathbf{n}_\mathbf{x}$. The *convolution* $C = A \star B$ is defined as the set of points obtained by adding the points of A and B having parallel normals (called *coherent points*), i.e.,

$$C = \{\mathbf{a} + \mathbf{b} : \mathbf{a} \in A, \mathbf{b} \in B, \mathbf{n}_\mathbf{a} \parallel \mathbf{n}_\mathbf{b}\}. \tag{1}$$

This notion is closely related to Minkowski sums. In the general case, the boundary of the Minkowski sum of two sets A, B is contained in the convolution of the two boundary surfaces $\delta A, \delta B$. In the case of convex sets, the boundary of the Minkowski sum and the convolution surface are identical. If one of the surfaces is a sphere, then the convolution is a two–sided offset surface. See e.g. [27] for more information and related references.

In what follows we assume that A and B are irreducible rational hypersurfaces. In this case, the convolution hypersurface $C = A \star B$ is algebraic but not necessarily rational and can be reducible or irreducible.

Let us consider two given regular rational parametric hypersurfaces $\mathbf{a}(u_1, \ldots, u_n)$, $\mathbf{b}(s_1, \ldots, s_n)$ in $\mathbb{R}^{n+1}$. For the sake of brevity, we will denote $\bar{u} = (u_1, \ldots, u_n)$, $\bar{s} = (s_1, \ldots, s_n)$, etc. In this case the normal direction is given simply as the unique direction perpendicular to all partial derivative vectors. The convolution formula then becomes

$$A \star B = \left\{ \mathbf{a}(\bar{u}) + \mathbf{b}(\bar{s}) : \operatorname{span}\left(\frac{\partial \mathbf{a}}{\partial u_1}(\bar{u}), \ldots, \frac{\partial \mathbf{a}}{\partial u_n}(\bar{u}) \right) = \operatorname{span}\left(\frac{\partial \mathbf{b}}{\partial s_1}(\bar{s}), \ldots, \frac{\partial \mathbf{b}}{\partial s_n}(\bar{s}) \right) \right\}.$$

The problem of finding mutually corresponding parametric values $\bar{u}$, $\bar{s}$ is highly nonlinear. In practice, the computation of the exact convolution is too complicated and various approximation algorithms are used instead, see e.g. [9]. In some cases, however, it is possible to find parameterizations of the given hypersurfaces which make the convolution computation trivial, as described by the following definition.

Definition 1. *Rational parameterizations* $\mathbf{a}(\bar{t})$, $\mathbf{b}(\bar{t})$ *of a given pair of hypersurfaces* $A, B \subset \mathbb{R}^{n+1}$ *over the same parameter domain* $\Omega \subset \mathbb{R}^n$ *will be called* coherent *iff the convolution condition (parallel normals) is automatically satisfied for every parameter value, i.e.,*

$$\operatorname{span}\left(\frac{\partial \mathbf{a}}{\partial t_1}(\bar{t}),\ldots,\frac{\partial \mathbf{a}}{\partial t_n}(\bar{t})\right) = \operatorname{span}\left(\frac{\partial \mathbf{b}}{\partial t_1}(\bar{t}),\ldots,\frac{\partial \mathbf{b}}{\partial t_n}(\bar{t})\right).$$

If we have coherent parameterizations of hypersurfaces (see Fig. 2) then by simply taking $\mathbf{a}(\bar{t}) + \mathbf{b}(\bar{t})$ we obtain a parameterization of the component of the convolution.[1]

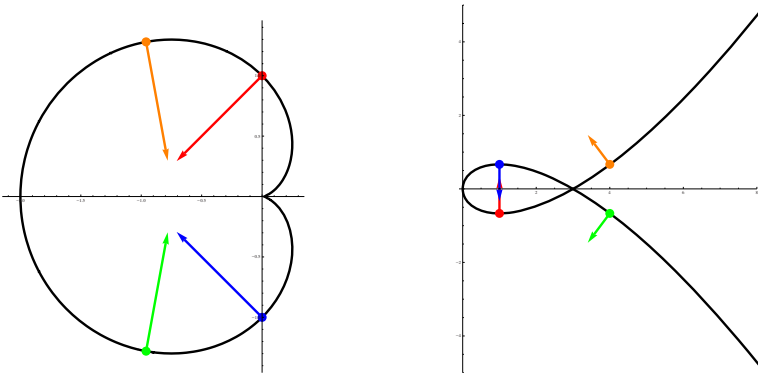


Fig. 1. The cardioid parameterized by $\mathbf{a}(u) = \left(\frac{2u^2-2u^4}{u^4+2u^2+1}, -\frac{4u^3}{u^4+2u^2+1}\right)^\top$ (left) and the Tschirnhausen cubic parameterized by $\mathbf{b}(s) = \left(s^2, s-\frac{s^3}{3}\right)^\top$ (right) with the normal vectors for $u,s = -2,-1,1,2$ (orange, red, blue, green). For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.

Coherent rational parameterizations do not exist for all given pairs of hypersurfaces. Clearly the rationality of the convolution, which is rather exceptional, is one of the necessary conditions. We want to treat the existence of coherent parameterizations with respect to given input parameterizations. More precisely, we will try to find the new

[1] In this paper we do not require from the coherent parameterizations that every pair of corresponding points is given by some parameter value. The question of existence for any $\mathbf{a} \in A$, $\mathbf{b} \in B$ with parallel normals $\mathbf{n_a} \parallel \mathbf{n_b}$ of some parameter value $\bar{t}_0 \in \Omega$ such that $\mathbf{a} = \mathbf{a}(\bar{t}_0)$, $\mathbf{b} = \mathbf{b}(\bar{t}_0)$ is related to the reducibility of the convolution and to the convolution degree of input hypersurfaces.

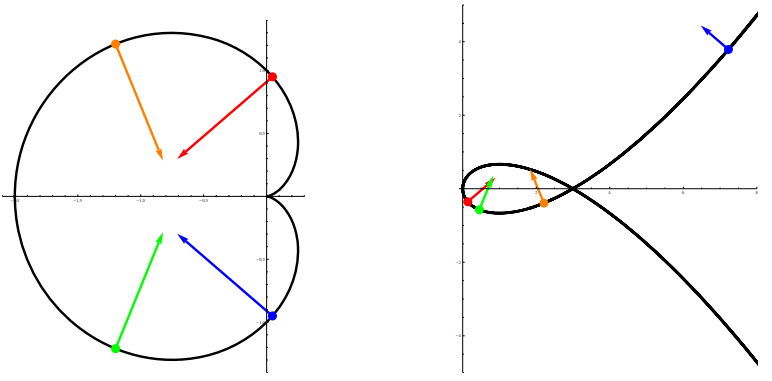


Fig. 2. The cardioid (left) and the Tschirnhausen cubic (right) parameterized by their coherent parameterizations $\tilde{\mathbf{a}}(t)$, $\tilde{\mathbf{b}}(t)$ (see Section 4) with the normal vectors for $t = -2/3, -2/5, 2/5, 2/3$ (orange, red, blue, green). For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.

parameterizations as (rational) reparameterizations of the given ones. As we will see later in this paper, in some cases it is sufficient to reparameterize only one of the input hypersurfaces and in other cases both parameterizations must be changed.

3 Changing Parameterization with Respect to a Given Normal Field

In this section we will present an algorithm which will be used in all the reparameterizations and which is based on the concept of a (extended) convolution ideal. The Gröbner bases of these ideals give an immediate answer to the coherent parameterization problem.

Let $\mathbf{a}(\bar{u})$ be a rational parameterization of a hypersurface A, $\alpha_j(\bar{u})$ relatively prime polynomial coordinates of normal vectors associated with $\mathbf{a}(\bar{u})$ and $\mathbf{n} = (n_1, \ldots, n_{n+1})^\top$ the formal variables for a (not necessarily unit) parametric normal field. At this moment we do not need any particular expressions for components of $\mathbf{n}$ and they are considered only as symbols into which particular rational functions of $\bar{t}$ will be substituted later. Now, we are looking for a new parameterization of A, so that the given $\mathbf{n}$ is a normal vector field associated with this new parameterization. This problem leads to a system of polynomial equations

$$\alpha_j(\bar{u}) = \lambda \cdot n_j, \lambda \neq 0, \quad j = 1, \ldots, n+1, \tag{2}$$

which is solved for $\bar{u}$, λ. Hence, this polynomial system can be treated using the Gröbner bases theory. Let us define the following ideals

$$I = \langle \alpha_1(\bar{u}) - \lambda n_1, \ldots, \alpha_{n+1}(\bar{u}) - \lambda n_{n+1}, 1 - w\lambda \rangle \subset k(n_1, \ldots, n_{n+1})[w, \bar{u}, \lambda], \tag{3}$$

which results from (2), and

$$\tilde{I} = \langle I, \text{numerator}(x_1 - a_1(\bar{u})), \ldots, \text{numerator}(x_{n+1} - a_{n+1}(\bar{u}))\rangle \subset \\ \subset k(n_1, \ldots, n_{n+1})[w, \bar{u}, \lambda, x_1, \ldots, x_{n+1}], \tag{4}$$

which also involves the parametric representation $\mathbf{a}(\bar{u})$ of A. The notation $k(a,b)[x,y]$ stands for the ring of polynomials in variables x, y with coefficients from the field of rational functions in a, b over an infinite field k. Then, I is called a *convolution ideal* and $\tilde{I}$ an *extended convolution ideal* of the hypersurface A parameterized by $\mathbf{a}(\bar{u})$. The convolution ideals were thoroughly studied in [16] but the extended convolution ideal is a new concept introduced in this paper. In what follows, we assume that the ideal I is zero-dimensional (and then consequently $\tilde{I}$ is also zero-dimensional) which is true for all parameterized hypersurfaces with non-degenerated Gauss image.

Next, we compute the reduced Gröbner basis $G_I = \{g_0, g_1, \ldots, g_{n+1}\}$ for the ideal I with respect to the lexicographic order for $w > u_1 > \ldots > u_n > \lambda$ and the reduced Gröbner basis $G_{\tilde{I}} = \{\tilde{g}_0, \tilde{g}_1, \ldots, \tilde{g}_{2n+2}\}$ for the ideal $\tilde{I}$ with respect to the lexicographic order for $w > u_1 > \ldots > u_n > \lambda > x_1 > \ldots > x_{n+1}$.[2] The product δ of the exponents of leading monomials of all polynomials g_j in G_I is called the *convolution degree* of the parameterization $\mathbf{a}(\bar{u})$, cf. [16]. The convolution degree δ depends on $\mathbf{a}(\bar{u})$. For different parameterizations of A we can obtain different convolution degrees.

Remark 2. One can also give a definition of the convolution degree of a hypersurface A denoted by Δ which is a geometrical invariant of A, i.e., it is independent from a particular parameterization of A. Then, Δ is less than or equal to δ for an arbitrary parameterization of A.

The Gröbner bases G_I and $G_{\tilde{I}}$ give an immediate answer to the coherent parameterization problem. The rational parameterization of A corresponding to $\mathbf{n}(\bar{t})$ exists if and only if (after substituting particular rational functions $n_j(\bar{t})$) the last $n+1$ polynomials of $G_{\tilde{I}}$ can be solved for $x_1, \ldots, x_{n+1}$ as rational functions in $\bar{t}$. It directly follows:

Lemma 3. *If the exponents of leading monomials of the last $n+1$ polynomials of $G_{\tilde{I}}$ are equal to 1, a coherent parameterization can be found for any input normal field $\mathbf{n}(\bar{t})$.*

In a similar way, a coherent parameterization can be found via a reparameterization $\bar{u} = \phi(\bar{t})$ if and only if the polynomials $g_1, \ldots, g_{n+1}$ from G_I can be solved for $u_1, \ldots, u_n$ as rational functions in $\bar{t}$. Moreover, if $\delta = 1$ then the last $n+1$ polynomials of $G_{\tilde{I}}$ can be solved for $x_1, \ldots, x_{n+1}$ as rational functions in $\bar{t}$. Hence, we obtain:

Lemma 4. *Let $\mathbf{a}(\bar{u})$ be a parameterized hypersurface in $\mathbb{R}^{n+1}$. If $\delta = 1$ then a coherent parameterization can be found for any input normal field $\mathbf{n}(\bar{t})$.*

Parameterizations with the convolution degree equal to one ($\delta = 1$) are the most suitable parameterizations with respect to the operation of convolution. These are called

[2] For detailed explanation, why G_I and $G_{\tilde{I}}$ contain $n+2$ and $2n+3$ polynomials, respectively, see [16].

GRC parameterizations and they provide rational convolutions (and thus also coherent parameterizations) generally with an arbitrary rational hypersurface. Nevertheless, also parameterizations of higher convolution degrees, so-called δ*-SRC parameterizations*, can provide coherent parameterizations and rational convolutions in the case all *RC conditions* (conditions preserving Rational Convolutions) induced by the given δ-SRC parameterization are fulfilled, see [16,17].

Several examples on application of convolution ideals can be found in [16]. The following example demonstrates the case when the convolution $A \star B$ is always rational but the rational reparameterization ϕ of $\mathbf{a}(\bar{u})$ such that $\mathbf{a}(\phi(\bar{s}))$, $\mathbf{b}(\bar{s})$ are coherent exists only for special parameterizations. In this case, the usage of extended convolution ideals is suitable. Typically, this happens when $\mathbf{a}(\bar{u})$ is not a *proper parameterization*[3].

Example 5. Let A be a paraboloid parameterized by $\mathbf{a}(u,v) = (u^2, v^2, u^4+v^4)^\top$. The corresponding convolution ideal I and extended convolution ideal $\tilde{I}$ are

$$I = \left\langle -8u^3v - n_1\lambda, -8uv^3 - n_2\lambda, +4uv - n_3\lambda, 1 - \lambda w \right\rangle, \tag{5}$$

$$\tilde{I} = \left\langle -8u^3v - n_1\lambda, -8uv^3 - n_2\lambda, +4uv - n_3\lambda, 1 - \lambda w, x - u^2, y - v^2, z - u^4 - v^4 \right\rangle. \tag{6}$$

The reduced Gröbner bases of these ideals are

$$G_I = \left\{ w - \frac{n_3^4\lambda}{4n_1n_2}, u + \frac{\lambda v n_3^2}{2n_2}, v^2 + \frac{n_2}{2n_3}, \lambda^2 - \frac{4n_1n_2}{n_3^4} \right\}, \tag{7}$$

$$G_{\tilde{I}} = \left\{ w - \frac{n_3^4\lambda}{4n_1n_2}, u + \frac{\lambda v n_3^2}{2n_2}, v^2 + \frac{n_2}{2n_3}, \lambda^2 - \frac{4n_1n_2}{n_3^4}, x + \frac{n_1}{2n_3}, y + \frac{n_2}{2n_3}, z - \frac{n_1^2+n_2^2}{4n_3^2} \right\}. \tag{8}$$

From (7) it follows that a rational reparameterization ϕ can be obtained only for exceptional surfaces $\mathbf{b}(s,t)$ (e.g. in case when the associated coordinates of the normal field are squares of some rational functions). On the other hand, the last three polynomials of $G_{\tilde{I}}$ indicate that the coherent parameterization can be found for any surface $\mathbf{b}(s,t)$ and its associated normal field $(n_1,n_2,n_3)^\top(s,t)$ (cf. Lemma 3). This is obviously true, since we know that the paraboloid is an LN surface. Hence, substituting the normal field of an arbitrary surface into (8) we do not obtain (in general) the rational reparameterization ϕ but we always obtain suitable coherent parameterization of the paraboloid.

4 Changing Parameterizations with Respect to Convolution

The main idea of the approach presented in Section 3 is based on reparameterizing the first hypersurface with respect to the normal field of the other hypersurface and/or

[3] Generally, a rational parameterization of a hypersurface provides a rational mapping from the affine or projective hyperplane onto this hypersurface. A parameterization is called *proper* if this mapping is birational, i.e., it has a rational inverse.

vice-versa. However, this direct method gives only a sufficient condition for deciding the rationality of the convolution and it can fail for some input parametric hypersurfaces $\mathbf{a}(\bar{u}), \mathbf{b}(\bar{s})$ despite the fact that coherent parameterizations exist. In this section, we introduce a new method which exceeds this limitation. The most important cases are $n = 1, 2$, so we will assume that in the sequel of this paper.

Example 6. Let A be the cardioid given by $\mathbf{a}(u) = \left(\frac{-2u^4+2u^2}{u^4+2u^2+1}, \frac{-4u^3}{u^4+2u^2+1}\right)^\top$ and B be the Tschirnhausen cubic parameterized by $\mathbf{b}(s) = \left(s^2, s - \frac{1}{3}s^3\right)^\top$. Applying the approach from Section 3 we arrive at $G_{I_\mathbf{a}}$, $G_{I_\mathbf{b}}$:

$$G_{I_\mathbf{a}} = \left\{ w + \frac{\lambda^2 n_2^3}{64} + \frac{(-27n_1^2 - 15n_2^2)\lambda}{64} + \frac{3n_2}{4}, u - \frac{\lambda^2 n_2^3}{72n_1} + \frac{(7n_2^2 + 27n_1^2)\lambda}{72n_1} + \right.$$
$$\left. + \frac{n_2}{9n_1}, \lambda^3 + \frac{(-27n_1^2 - 15n_2^2)\lambda^2}{n_2^3} + \frac{48\lambda}{n_2^2} + \frac{64}{n_2^3} \right\}, \tag{9}$$

$$G_{I_\mathbf{b}} = \left\{ w - \frac{\lambda n_2^2}{4} - n_1, s + \frac{\lambda n_2}{2}, \lambda^2 + \frac{4\lambda n_1}{n_2^2} - \frac{4}{n_2^2}, \right\}. \tag{10}$$

To summarize, $\mathbf{a}(u)$ is 3-SRC parameterization and $\mathbf{b}(s)$ is 2-SRC parameterization with mutually incompatible RC conditions. We compute bases $G_{\tilde{I}_\mathbf{a}}$, $G_{\tilde{I}_\mathbf{b}}$:

$$G_{\tilde{I}_\mathbf{a}} = \{w + p_1 y^2 + p_2 y + p_3, u + p_4 y^2 + p_5 y + p_6, \lambda + p_7 y^2 + p_8 y + p_9,$$
$$x + p_{10} y^2 + p_{11} y + p_{12}, y^3 + p_{13} y^2 + p_{14} y + p_{15}\}, \tag{11}$$
$$G_{\tilde{I}_\mathbf{b}} = \{w + q_1 y + q_2, s + q_3 y + q_4, \lambda + q_5 y + q_6, x + q_7 y + q_8,$$
$$y^2 + q_9 y + q_{10}\}, \tag{12}$$

where p_i, q_j are certain more complicated rational functions in n_1, n_2. Substituing the normal fields $\mathbf{n_b} = \left(1 - s^2, 2s\right)^\top$, $\mathbf{n_a} = \left((u^2 - 3)u, 3u^2 - 1\right)^\top$ into the Gröbner bases $G_{\tilde{I}_\mathbf{a}}$, $G_{\tilde{I}_\mathbf{b}}$, respectively, we get from the last two polynomials of each basis two systems of equations in x, y, however in both cases with non-rational solutions. Thus the direct method presented in Section 3 fails.

As we will see later, for some pairs of rational hypersurfaces (e.g. for curves in Example 6), it is possible to obtain a suitable normal field directly from their convolution and then parameterize them coherently with respect to this field.

For this purpose we apply a generalization of the so-called *support function representation of hypersurfaces* (cf. [1,11,29]). The support function is a tool most widely used in the convex geometry for the representation of convex bodies, see e.g. [12]. Recently, this concept has been extended to the so-called quasi-convex hypersurfaces in [29].

A hypersurface in $\mathbb{R}^{n+1}$ is described as the envelope of its tangent hyperplanes

$$T_\mathbf{n} := \{\mathbf{x} : \mathbf{n} \cdot \mathbf{x} = h(\mathbf{n})\}, \tag{13}$$

where the *support function (SF)* $h(\mathbf{n})$ is a function defined on the unit sphere S_n (or a suitable subset). Among others, this representation is very suitable for describing

convolutions of hypersurfaces as this operation corresponds to the sum of the associated support functions, i.e.,

$$h_C = h_A + h_B. \tag{14}$$

However, given a parametric representation of a hypersurface, it is not always possible to represent it via SF – mainly due to the fact, that for each vector $\mathbf{n}$ only one value of h is possible.

For this reason we use a more general representation which is available for all algebraic hypersurfaces with non-degenerated Gauss image. A hypersurface is here represented as an envelope of tangent hyperplanes (13) where $\mathbf{n}$ and h satisfy the homogeneous polynomial equation

$$D(\mathbf{n},h) = 0. \tag{15}$$

Recall that $D(\mathbf{n},h) = 0$ is a *dual representation* of a hypersurface. To emphasize the relation to the standard one-valued SF representation and multi-valued SF representation (15), we will call $D(\mathbf{n},h) = 0$ the *implicit support function* (or shortly *ISF*) *representation* of a hypersurface. More precisely, we mean by ISF the function $\mathbf{n} \mapsto h(\mathbf{n})$ (cf. (13)) which is implicitly defined by (15), where $\mathbf{n}$ is assumed to be a unit vector – for further details see [1], where the theory of implicitly given support functions is discussed. Obviously, the dual representation (15) does not generally require the normalized normal vectors $\mathbf{n}$. On the contrary, SF is a function defined only on the unit sphere S_n.

For a parametrically or implicitly given hypersurface we can find its ISF using a suitable elimination technique – see e.g. [1, 11]. Next, we recall the following well-known facts from algebraic geometry:

Lemma 7. *For the dual hypersurface $D(\mathbf{n},h) = 0$ of a hypersurface A, it holds:*

1. *A is irreducible if and only if $D(\mathbf{n},h)$ is irreducible.*
2. *There exists a rational representation of A if and only if the zero locus of the corresponding $D(\mathbf{n},h) = 0$ is rational.*

Remark 8. Let us emphasize that the rationality of the dual hypersurface $D(\mathbf{n},h) = 0$ does not guarantee the existence of a rational parameterization of the corresponding ISF, which is a simultaneous rational parameterization of (15) and of $\mathbf{n} \in S_n$. Hence, since $\mathbf{n}$ is assumed to be a unit vector for ISF, the rationality of ISF is fulfilled only for primal hypersurfaces with Pythagorean normals.

Given two ISF representations D_A, D_B of A, B we can easily compute the ISF representation D_C of the convolution hypersurface $C = A \star B$. It is enough to eliminate variables h_A, h_B from the system of equations

$$\begin{aligned} D_A(\mathbf{n},h_A) &= 0, \\ D_B(\mathbf{n},h_B) &= 0, \\ h_C - h_A - h_B &= 0. \end{aligned} \tag{16}$$

As mentioned in Section 2, the convolution hypersurface $C = A \star B$ can be reducible, or irreducible. Firstly, we exclude from our further considerations components which are degenerated (i.e., with dimensions less than n). Furthermore, to distinguish between

different kinds of irreducible non-degenerated components, we formulate the following definition:[4]

Definition 9. *A non-degenerated irreducible component C_0 of the convolution hypersurface $C = A \star B$ is called* simple *if there exists a dense set $\mathscr{S} \subset C_0$ such that every $\mathbf{c} \in \mathscr{S}$ is generated by exactly one pair $(\mathbf{a},\mathbf{b})$, $\mathbf{a} \in A$, $\mathbf{b} \in B$. Otherwise C_0 is called* special.

In the case of reducible convolution hypersurfaces (and hence reducible D_C) we speak analogously about *simple* and *special components* of the corresponding dual hypersurface described by the implicit support function D_C.

Remark 10. Analogously to [2] we can formulate:

1. Every $C = A \star B$ has at least one simple component.
2. If $C = A \star B$ is irreducible then it is simple.
3. Special components of convolution hypersurfaces are typically those ones when $C = (A \star B) \star B$ for centrally symmetric B – e.g. in case when offsets to offsets are constructed.
4. Simple components of C correspond to simple components of the associated dual hypersurface described by $D_C = 0$.

There are efficient algorithms to check rationality of (a component of) $D_C = 0$ and, if possible, to construct its rational parameterization $\mathbf{n}(\bar{t}), h_C(\bar{t})$ – see e.g. [24, 25, 30, 31]. The normal field $\mathbf{n}(\bar{t})$ can be then used to obtain coherent parameterizations of the input surfaces (as shown in the continuation of our example).

Example 6 (continued). Now we use the approach based on the dual representation. First, we compute

$$D_A(n_1,n_2,h_A) = 16h_A^3 + 24n_1h_A^2 - 27n_2^2h_A - 15n_1^2h_A + 2n_1^3, \tag{17}$$
$$D_B(n_1,n_2,h_B) = 9h_B^2n_2^2 - 12h_Bn_1^3 - 18h_Bn_1n_2^2 - 3n_1^2n_2^2 - 4n_2^4. \tag{18}$$

Next, by eliminating h_A, h_B from (16) we obtain the dual representation of $C = A \star B$ in the form

$$\begin{aligned} D_C(n_1,n_2,h_C) = {} & (186624n_2^6)h_C^6 + (-559872n_1n_2^6 - 746496n_1^3n_2^4)h_C^5 + \\ & + (-878688n_2^8 - 676512n_2^6n_1^2 + 1119744n_2^4n_1^4 + 995328n_2^2n_1^6)h_C^4 + \\ & + (-442368n_1^9 + 2072304n_2^8n_1 + 4944240n_2^6n_1^3 + 3297024n_2^4n_1^5)h_C^3 + \\ & + (-663552n_1^{10} - 3504384n_1^8n_2^2 - 4316247n_1^4n_2^6 - 6453216n_1^6n_2^4 - \\ & - 63990n_1^2n_2^8 + 642033n_2^{10})h_C^2 + (414720n_1^{11} + 1938816n_1^9n_2^2 - \\ & - 753570n_1n_2^{10} + 772362n_1^5n_2^6 + 2832948n_1^7n_2^4 - 1290204n_1^3n_2^8)h_C + \\ & + (289224n_1^8n_2^4 + 458010n_1^4n_2^8 - 108387n_1^2n_2^{10} - 82944n_1^{10}n_2^2 + \\ & + 755109n_1^6n_2^6 - 55296n_1^{12} - 128164n_2^{12}). \end{aligned} \tag{19}$$

[4] This definition is motivated by [2] where the defined notions were introduced for reducible offset hypersurfaces.

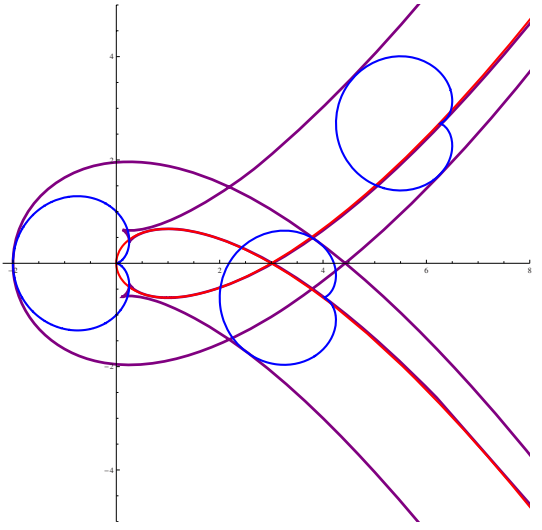


Fig. 3. The convolution curve C (purple) of the cardioid A (blue) and the Tschirnhausen cubic B (red). For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.

D_C is irreducible, i.e., C is simple. It can be shown that $\text{genus}(D_C) = 0$ and thus D_C (and also C) is a rational curve. We use a parameterization algorithm (see [25, 30, 31]) and obtain

$$n_1(t) = \frac{6t - 20t^3 + 6t^5}{t^6 + 3t^4 + 16t^3 + 3t^2 + 1}, \quad n_2(t) = \frac{1 - 15t^2 + 15t^4 - t^6}{t^6 + 3t^4 + 16t^3 + 3t^2 + 1}, \tag{20}$$

$$h_C(t) = -\frac{2}{3}\frac{1-3t-54t^5-23t^9-54t^7-3t^{11}+264t^6-23t^3-9t^4-12t^2-9t^8-12t^{10}+t^{12}}{1+14t^3+t^{12}+14t^9+6t^{11}+6t-300t^6+6t^{10}+111t^8+12t^7+111t^4+12t^5+6t^2}. \tag{21}$$

Now applying the approach from Section 3 we obtain the following rational reparameterizations $u = \phi_i(t)$ and $s = \psi_j(t)$

$$\phi_1 : u = \frac{2t}{1-t^2}, \quad \phi_2 : u = \frac{t^4\sqrt{3} - 8t^3 + 2\sqrt{3}t^2 + 8t + \sqrt{3}}{t^4 - 14t^2 + 1}, \tag{22}$$

$$\phi_3 : u = -\frac{t^4\sqrt{3} + 8t^3 + 2\sqrt{3}t^2 - 8t + \sqrt{3}}{t^4 - 14t^2 + 1},$$

and

$$\psi_1 : s = -\frac{t^3 + 3t^2 - 3t - 1}{t^3 - 3t^2 - 3t + 1}, \quad \psi_2 : s = \frac{t^3 - 3t^2 - 3t + 1}{t^3 + 3t^2 - 3t - 1}. \tag{23}$$

An arbitrary pair (ϕ_i, ψ_j), $i = 1,2,3$, $j = 1,2$, yields coherent parameterizations $\mathbf{a}(\phi_i(t))$, $\mathbf{b}(\psi_j(t))$ of the cardioid and the Tschirnhausen cubic, respectively, e.g. for ϕ_1 and ψ_1 we have

$$\tilde{\mathbf{a}}(t) = \left(\frac{8t^2(-6t^2+t^4+1)}{(t^2+1)^4}, \frac{32(t^2-1)t^3}{(t^2+1)^4} \right)^\top, \tag{24}$$

$$\tilde{\mathbf{b}}(t) = \left(\frac{(t^3+3t^2-3t-1)^2}{(t^3-3t^2-3t+1)^2}, \right.$$
$$\left. \frac{-2(t^3+3t^2-3t-1)(t^6-12t^5+3t^4+40t^3+3t^2-12t+1}{3(t^3-3t^2-3t+1)^3} \right)^\top. \tag{25}$$

Obviously, (20) is not a rational parameterization of ISF since $\mathbf{n}$ is not a unit vector. However, it can be easily found if we divide it by $|\mathbf{n}|$ which is rational in this case.

The success of the previous example is by no means exceptional. In fact if the convolution is rational, then the two original hypersurfaces can be parameterized coherently with respect to a normal field (of a simple component) of the convolution.

Theorem 11. *Let $\mathbf{a}(\bar{u})$ and $\mathbf{b}(\bar{s})$ be parameterizations of the rational hypersurfaces A and B, respectively. Let $D_C(\mathbf{n}, h_C)$ be the dual representation of the convolution hypersurface $C = A \star B$. If $(\mathbf{n}(\bar{t}), h_C(\bar{t}))^\top$ is a rational parameterization of a simple component of $D_C(\mathbf{n}, h_C)$, then there exist parameterizations $\mathbf{a}(\bar{t})$ and $\mathbf{b}(\bar{t})$ of A, B which are coherent with respect to the normal field $\mathbf{n}(\bar{t})$ and therefore with respect to each other.*

Proof. In general, the convolution hypersurface C can be written in the form $C = C_1 \cup C_2 \cup \ldots \cup C_l$, where C_i can be degenerated, simple, or special components. Let us assume that C_1 is a simple component and let $D_{C_1}(\mathbf{n}, h_C)$ be a factor of $D_C(\mathbf{n}, h_C)$ associated to the component C_1. Then we can construct a set of equivalence classes

$$\mathscr{M} = \left\{ (\mathbf{n} : h_C : h_A : h_B) \in k^{n+1} \times k \times k \times k \;\middle|\; \begin{array}{l} D_A(\mathbf{n}, h_A) = 0 \\ D_B(\mathbf{n}, h_B) = 0 \\ D_C(\mathbf{n}, h_C) = 0 \end{array} \right\}$$
$$= \mathscr{M}_1 \cup \mathscr{M}_2 \cup \ldots \cup \mathscr{M}_l. \tag{26}$$

Further, we consider the natural projections

$$\pi_2 : \mathscr{M} \to \mathscr{V}(D_C) : (\mathbf{n} : h_C : h_A : h_B) \mapsto (\mathbf{n} : h_C),$$
$$\pi_3 : \mathscr{M} \to \mathscr{V}(D_A) : (\mathbf{n} : h_C : h_A : h_B) \mapsto (\mathbf{n} : h_A),$$
$$\pi_4 : \mathscr{M} \to \mathscr{V}(D_B) : (\mathbf{n} : h_C : h_A : h_B) \mapsto (\mathbf{n} : h_B),$$

where $\mathscr{V}(D_A)$ is a projective variety given by D_A, etc. All natural projections π_2, π_3, π_4 are rational maps. Since C_1 is simple, i.e., also the associated $\mathscr{V}(D_{C_1})$ is simple, $\pi_2 : \mathscr{M}_1 \to \mathscr{V}(D_{C_1})$ is 1:1. This implies that $\pi_2 : \mathscr{M}_1 \to \mathscr{V}(D_{C_1})$ is a birational map. However, π_3 and π_4 do not have to be 1:1. In general, π_3 is $\deg_{h_B}(D_B(\mathbf{n}, h_B)) : 1$ and $\deg_{h_A}(D_A(\mathbf{n}, h_A)) : 1$. Thus, π_3 and π_4 are only rational maps.

Hence, we obtain two maps

$$\xi : \mathscr{V}(D_{C_1}) \to \mathscr{V}(D_A) : (\mathbf{n} : h_C) \mapsto (\mathbf{n} : h_A),$$
$$\zeta : \mathscr{V}(D_{C_1}) \to \mathscr{V}(D_B) : (\mathbf{n} : h_C) \mapsto (\mathbf{n} : h_B),$$

where $\xi = \pi_3 \circ \pi_2^{-1}$ and $\zeta = \pi_4 \circ \pi_2^{-1}$, i.e., ξ and ζ are rational. This implies that any rational parameterization of D_C provides rational parameterizations of D_A and D_B. □

Example 12. Let us consider two rational hypersurfaces A and B parameterized by $\mathbf{a}(\bar{u})$ and $\mathbf{b}(\bar{s})$, respectively, with rational convolution $C = A \star B$. However, there exist only non-rational dependencies $\bar{u} = \phi(\bar{s})$, and $\bar{s} = \psi(\bar{u})$ yielding parallel normal fields. We can apply the approach from Example 6, compute a suitable rational parameterization $\mathbf{n}(\bar{t})$ from $D_C(\mathbf{n}, h_C)$ and then set it to polynomials in Gröbner bases of the extended convolution ideals $\tilde{I}_\mathbf{a}$, $\tilde{I}_\mathbf{b}$. As in Example 5, it is not enough to use the ideals $I_\mathbf{a}$, $I_\mathbf{b}$, because they do not give rational reparameterizations. Typically, this is the case when both parameterizations $\mathbf{a}(\bar{u})$, $\mathbf{b}(\bar{s})$ are not *proper parameterizations*.

5 Summary and Conclusion

Finally, we will sketch the general algorithm for computing coherent parameterizations based on the previous results – see Algorithm 1.

Algorithm 1. Finds coherent parameterizations of given parametric hypersurfaces (if they exist).

Input: Hypersurfaces A, B given by their parameterizations $\mathbf{a}(\bar{u})$ and $\mathbf{b}(\bar{s})$.
Output: Coherent parameterizations of A and B or "Convolution is not rational".

1: Try to rationally reparameterize $\mathbf{a}(\bar{u})$ with respect to the normal field $\mathbf{n}_\mathbf{b}(\bar{s})$ of B. For this purpose compute Gröbner basis $G_{I_\mathbf{a}}$ of the convolution ideal of $\mathbf{a}(\bar{u})$. Substitute $\mathbf{n} = \mathbf{n}_\mathbf{b}(\bar{s})$ in this basis and try to solve it for $\bar{u}$, obtaining rational parameterization $\mathbf{a}(\phi(\bar{s}))$ coherent with $\mathbf{b}(\bar{s})$. If necessary, repeat Step 1 with $\mathbf{a}$ and $\mathbf{b}$ swapped.
2: If Step 1 fails, try to find a new parameterization of A coherent with the normal field $\mathbf{n}_\mathbf{b}(\bar{s})$ of B. For this purpose compute Gröbner basis $G_{\tilde{I}_\mathbf{a}}$ of the extended convolution ideal of A. Substitute $\mathbf{n} = \mathbf{n}_\mathbf{b}(\bar{s})$ in this basis and try to solve it for $\mathbf{x}$, obtaining new parameterization of A coherent with $\mathbf{b}(\bar{s})$. If necessary, repeat Step 2 with $\mathbf{a}$ and $\mathbf{b}$ swapped.
3: If Step 2 fails, compute successively the ISFs of A, B and of their convolution $A \star B$. If the dual of the convolution is not rational, then return "Convolution is not rational". Otherwise compute rational parametrization $(\mathbf{n}(\bar{t}), h(\bar{t}))$ of its simple component. Substitute $\mathbf{n} = \mathbf{n}(\bar{t})$ in $G_{I_\mathbf{a}}$ and $G_{I_\mathbf{b}}$ and try to solve them for $\bar{u}$ and $\bar{s}$, respectively, obtaining coherent parameterizations $\mathbf{a}(\phi(\bar{t}))$ and $\mathbf{b}(\psi(\bar{t}))$.
4: If Step 3 fails, substitute $\mathbf{n} = \mathbf{n}(\bar{t})$ in $G_{\tilde{I}_\mathbf{a}}$, $G_{\tilde{I}_\mathbf{b}}$ respectively and solve each of them for $\mathbf{x}$ obtaining new coherent parameterizations of A and B.

Details of Steps 1 and 2 are explained in Section 3 and Example 5 while details of Steps 3 and 4 are explained in Section 4 and Example 6. Theorem 11 ensures that coherent parameterizations are found by the algorithm, provided that the convolution is rational. The examples given in this paper show that for various input data the algorithm can stop at any step successfully providing coherent parameterizations.

Our approach to the problem of finding coherent parameterizations of given parametric hypersurfaces is an "economic" one – we try to find the simplest possible way to coherent parameterizations. Thus, if possible, one of the input parameterizations is kept unchanged. Also, the new parameterizations are, if possible, found via rational reparameterization of existing ones. Using concepts of the extended convolution ideal and the

dual representation in connection with implicit support function, we were able to solve fully the problem and to describe an algorithm which finds coherent parameterizations (if they exist) in an optimal way.

In the future we plan to develop further the concepts of the extended convolution ideal and implicit support function and namely to study them in relation to the convolution degree of hypersurfaces. We also plan to apply these tools for the computation of convolutions of implicit algebraic hypersurfaces.

Acknowledgments

The authors were supported by the Research Plan MSM 4977751301. We thank to all referees for their comments which helped us to improve the paper.

References

1. Aigner, M., Jüttler, B., Gonzales-Vega, L., Schicho, J.: Parameterizing surfaces with certain special support functions, including offsets of quadrics and rationally supported surface. Journal of Symbolic Computation 44, 180–191 (2009)
2. Arrondo, E., Sendra, J., Sendra, J.R.: Parametric generalized offsets to hypersurfaces. Journal of Symbolic Computation 23(2-3), 267–285 (1997)
3. Becker, T., Weispfenning, V.: Gröbner bases – a computational approach to commutative algebra. Graduate Texts in Mathematics. Springer, New York (1993)
4. Farouki, R.T., Sakkalis, T.: Pythagorean hodographs. IBM Journal of Research and Development 34(5), 736–752 (1990)
5. Farouki, R.T., Sakkalis, T.: Pythagorean hodograph space curves. Advances in Computational Mathematics 2(1), 41–66 (1994)
6. Farouki, R.T., Neff, C.A.: Hermite interpolation by Pythagorean-hodograph quintics. Mathematics of Computation 64(212), 1589–1609 (1995)
7. Farouki, R.T.: Pythagorean-hodograph curves: Algebra and Geometry Inseparable. Geometry and Computing. Springer, Heidelberg (2008)
8. Farouki, R.T., Manni, C., Sestini, A.: Spatial C^2 PH quintic splines. In: Lynche, T., Mazure, M.L., Schumaker, L.L. (eds.) Curve and Surface Design: St. Malo 2002, pp. 147–156. Nashboro Press (2003)
9. Farouki, R.T., Hass, J.: Evaluating the boundary and covering degree of planar Minkowski sums and other geometrical convolutions. Journal of Computational and Applied Mathematics 209(2), 246–266 (2007)
10. Farouki, R.T., Sakkalis, T.: Rational space curves are not "unit speed". Computer Aided Geometric Design 24, 238–240 (2007)
11. Gravesen, J., Jüttler, B., Šír, Z.: On Rationally Supported Surfaces. Computer Aided Geometric Design 25(4-5), 320–331 (2008)
12. Gruber, P.M., Wills, J.M.: Handbook of convex geometry. North–Holland, Amsterdam (1993)
13. Jüttler, B.: Triangular Bézier surface patches with linear normal vector field. In: Cripps, R. (ed.) The Mathematics of Surfaces VIII, pp. 431–446. Information Geometers (1998)
14. Jüttler, B.: Hermite interpolation by Pythagorean hodograph curves of degree seven. Mathematics of Computation 70, 1089–1111 (2001)
15. Kosinka, J., Jüttler, B.: G^1 Hermite interpolation by Minkowski Pythagorean hodograph cubics. Computer Aided Geometric Design 23, 401–418 (2006)

16. Lávička, M., Bastl, B.: Rational hypersurfaces with rational convolutions. Computer Aided Geometric Design 24, 410–426 (2007)
17. Lávička, M., Bastl, B.: PN surfaces and their convolutions with rational surfaces. Computer Aided Geometric Design 25, 763–774 (2008)
18. Moon, H.P.: Minkowski Pythagorean hodographs. Computer Aided Geometric Design 16, 739–753 (1999)
19. Peternell, M., Pottmann, H.: A Laguerre geometric approach to rational offsets. Computer Aided Geometric Design 15, 223–249 (1998)
20. Peternell, M., Manhart, F.: The convolution of a paraboloid and a parametrized surface. Journal for Geometry and Graphics 7, 157–171 (2003)
21. Pottmann, H.: Rational curves and surfaces with rational offsets. Computer Aided Geometric Design 12, 175–192 (1995)
22. Pottmann, H., Peternell, M.: Applications of Laguerre geometry in CAGD. Computer Aided Geometric Design 15, 165–186 (1998)
23. Sampoli, M.L., Peternell, M., Jüttler, B.: Rational surfaces with linear normals and their convolutions with rational surfaces. Computer Aided Geometric Design 23, 179–192 (2006)
24. Schicho, J.: Rational parametrization of surfaces. Journal of Symbolic Computation 26, 1–29 (1998)
25. Sendra, J.R., Winkler, F.: Symbolic Parametrization of Curves. Journal of Symbolic Computation 12, 607–631 (1990)
26. Šír, Z., Feichtinger, R., Jüttler, B.: Approximating curves and their offsets using biarcs and Pythagorean hodograph quintics. Computer-Aided Design 38, 608–618 (2006)
27. Šír, Z., Gravesen, J., Jüttler, B.: Computing Convolutions and Minkowski sums via Support Functions. In: Chenin, P., Lyche, T., Schumaker, L.L. (eds.) Curve and Surface Design: Avignon 2006, pp. 244–253. Nashboro Press (2007)
28. Šír, Z., Jüttler, B.: C^2 Hermite interpolation by Pythagorean hodograph space curves. Mathematics of Computation 76, 1373–1391 (2007)
29. Šír, Z., Gravesen, J., Jüttler, B.: Curves and surfaces represented by polynomial support functions. Theoretical Computer Science 392(1-3), 141–157 (2008)
30. van Hoeij, M.: Rational parametrizations of algebraic curves using a canonical divisor. Journal of Symbolic Computation 23, 209–227 (1997)
31. Winkler, F.: Polynomial Algorithms in Computer Algebra. Texts and Monographs in Symbolic Computation. Springer, Wien (1996)

An Introduction to Guided and Polar Surfacing

Jörg Peters[1] and Kęstutis Karčiauskas[2]

[1] Department of Computer and Information Science and Engineering
University of Florida, USA
jorg@cise.ufl.edu
[2] Faculty of Mathematics and Informatics, Vilnius University, Lithuania
kestutis.karciauskas@mif.vu.lt

Abstract. This paper gives an overview of two recent techniques for high-quality surface constructions: polar layout and the guided approach. We demonstrate the challenge of high-quality surface construction by examples since the notion of surface quality lacks an overarching theory. A key ingredient of high-quality constructions is a good layout of the surface pieces. Polar layout simplifies design and is natural where a high number of pieces meet. A second ingredient is separation of shape design from surface representation by creating an initial guide shape and leveraging classical approximation-theoretic tools to construct a final surface compatible with industry standards, either as a finite number of polynomial patches or as a subdivision process. An example construction generating guided C^2 surfaces from patches of degree bi-3 highlights the power of the approach.

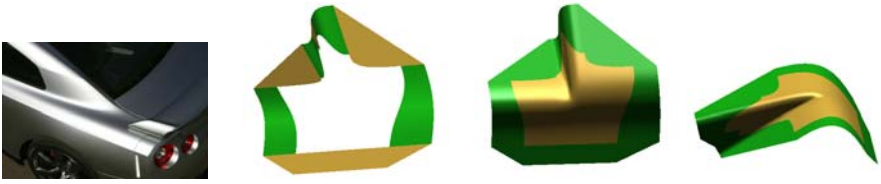
1 Introduction

Understanding what makes a surface 'fair' or 'of high-quality' and understanding the construction of curvature continuous, parametric surfaces well enough to no longer treat multi-sided blends as a difficult exception is an ongoing research challenge for the community. This paper gives an overview of two useful techniques for constructing high-quality surfaces. Such surfaces satisfy more stringent shape and differential requirements, outlined in Section 2, than might be needed for computer graphics (where speed of construction and rendering, with displacement for detail, is the main challenge) or result from scanning existing artifacts (where compression and manipulation are major challenges).

2 The Higher-Order Surfacing Challenge

The ability to efficiently design geometrically-high-quality surfaces has a direct impact on better product design and industrial competitiveness. Yet there has been little progress in automating high-quality surface generation. In practice, industry manages by time-consuming human intervention and by rejecting designs that are considered too complex. Regions where several primary surfaces meet are of special concern. Figure 1 shows a C-pillar configuration where a car roof support meets the trunk and fender region. As Figure 1 (b) illustrates, the primary surfaces (light, gold color in the example) are trimmed, then pairwise joined by blend surfaces (darker, green color) leaving a multi-sided hole to be filled in (see e.g. [1]). The quality of the multi-sided fill depends

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 299–315, 2010.



(a) (b) (c)

Fig. 1. Multi-sided blend. (a) Automobile C-pillar, (b) Trimmed primary surfaces pairwise blended. (c) High-quality fill.

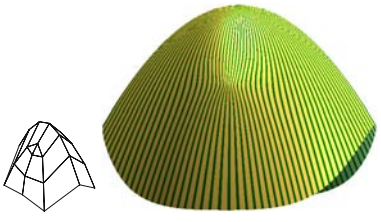
on (i) preserving feature lines and functional curves and (ii) avoiding extraneous dips in the surface so that highlight lines vary smoothly and monotonically almost everywhere. Some car designers additionally check for (iii) undue variation in the curvature by sliding their hand over a first physical prototype of the surface. The multi-sided surface blend of Figure 1 (c) was judged A-class (by a major car manufacturer) 'because our tests did not show anything undesirable'. Short of manufacturing the part, such tests consist of simulating the reflection of parallel light sources, normal and curvature distributions and section curvature (hedgehog) displays on the computer model.

Figure 2 shows some typical shape defects, detected in the computer model, of surfaces generated by standard and more sophisticated blend constructions in the literature. Some of theses defects are macroscopic, and immediately visible by computer rendering, while more subtle ones become evident only when applying a Gauss-curvature texture, i.e. color the surface according to Gauss curvature. Example (a) shows the effect of the finite support of the underlying basis functions: a ridge diagonal to the parameter lines of a tensor-product spline results in aliasing ripples. Example (b) shows a similar ripple formation for Loop subdivision [2], where low and high-valent vertices meet (the control net is that of a pinched cylinder and features a central point of valence 20 surrounded by points of valence 3.) Example (c) display a more subtle defect, visible as a pinch-point in the highlight lines. Example (d) illustrates a pervasive problem of first-generation subdivision algorithms for a range of input meshes: the shape is 'hybrid', i.e. each nested surface ring of a subdivision surface has both positive and negative Gauss curvature so that the curvature is not well-defined in the limit. This hybrid shape is visualized, for one surface ring, by overlaying on the Bézier net, curvature needles pointed up (red scale, positive Gauss curvature) and down (blue-green scale, negative Gauss curvature). Example (e) illustrates how 'tuning' Catmull-Clark subdivision [3] to achieve bounded curvature results in oscillation (of the Gauss curvature). The standard Catmull-Clark limit surface has extraordinary points (corresponding to non-4-valent control nodes) with infinite curvature and is generically hyperbolic. Example (f) shows unintended curvature oscillations (Gauss curvature shading) in a transition layer and not consistent with the monotone curvature variation at the center and in the surrounding outer region.

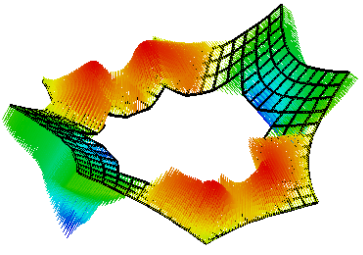
Some defects, certainly (a), can be avoided by the experienced designer (by aligning features with boundaries of the support of the basis functions). But [4, 5] showed that shape deficiency (d) is intrinsic to all first-generation subdivision surface constructions

(a) Aliasing ripples

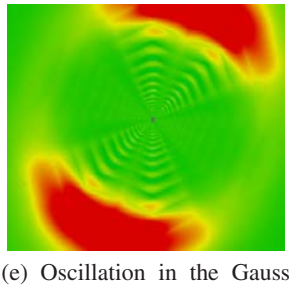
(b) Low-valence to high-valence transition ripples



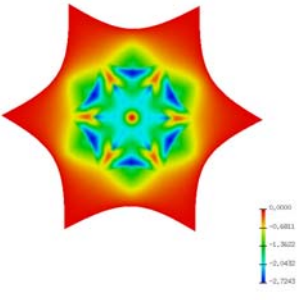
(c) Pinch-point in the highlight lines



(d) Hybrid shape made visible by signed Gauss curvature needles (blue-green downward pointing needles indicate negative Gauss curvature; red-green upward positive curvature)



(e) Oscillation in the Gauss curvature due to boundedness



(f) Oscillation in the Gauss curvature in a transition layer

Fig. 2. Surface **shape defects**

such as the popular Catmull-Clark subdivision [3]: While subdivision initially does a fine job in smoothing out transitions between primary surfaces, the shape problems are typically concentrated in the neighborhood of so-called extraordinary points. The surface quality deteriorates progressively as 'the wrong' eigenfrequencies take over near such a point, and force for example saddle shapes no matter how carefully a designer

might hand-adjust the input data. Since the shape defects manifest themselves already in the first steps, algorithms that build on or initialize with several steps of Catmull-Clark subdivision inherit coarse-level defects. More smoothing of the regular regions [6] appears to be even more detrimental to the shape near these extraordinary points. Conversely, constructions that do not make use of multiple refinement steps and construct the blend from a minimal number of polynomial pieces pay for the abrupt transition of patch type with curvature fluctuations (Figure 2(f)).

Since no overarching theory of surface 'fairness' exists to date, we extract from folklore, theory and many examples the following working definition of a high-quality surface blend. Given input data to be matched, a multi-sided surface blend is of *high quality* if *it does not create curvature features not implied by the input data*. That is, the blend surface should (a) minimize curvature variation but (b) preserve feature lines. By default, the blend should therefore be at least curvature continuous. In practice, since many downstream algorithms work directly on the parameterization, we also expect a high-quality surface blend to have (c) a good distribution of parameter lines.

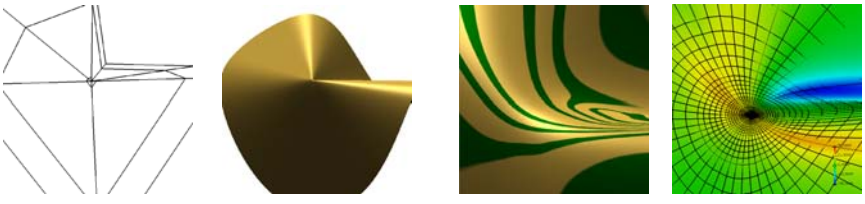


Fig. 3. Feature preservation: minimal curvature variation and parameter lines aligned with feature lines

One approach to improving surface quality is to optimize a (variational) functional. Shape optimization functionals include the thin plate energy [7, 8, 9, 10, 11, 12], the membrane energy [13], and total curvature [14, 15]. Replacing differential operators by differences, the optimization problems can be solved as sparse, finite linear or nonlinear systems in the coefficients of a chosen basis. *Evolution* formulations seek to reduce variation which, for a surface $\mathbf{x}(u,v)$, amounts to solving the (geometric diffusion) equation $\partial_t \mathbf{x} = \Delta_{\mathbf{x}} \mathbf{x}$ in terms of the $\mathbf{x}$-dependent Laplace-Beltrami operator $\Delta_{\mathbf{x}}$ [16]. Discretized versions of the Laplacian have been applied to polyhedral meshes, say in Taubin's pioneering paper [17], and, in the context of smoothing interpolatory subdivision and multiresolution editing [13, 18]. Desbrun et al. [19] use implicit discretization to stabilize numerical smoothing and Clarenz et al. introduced anisotropic geometric diffusion to enhance features while smoothing. Similar discretizations are based on [20, 21]. *Flow techniques* evolve the surface according to $\partial_t \mathbf{x} = \mathbf{v}(\mathbf{x},t)$ where $\mathbf{v}$ represents a velocity field [22, 23]. For example, geodesic curvature flow has been used for smoothing functions on surfaces [19, 24, 25]. Schneider and Kobbelt [26] distinguish between 'outer fairness' and 'inner fairness'. Outer fairness amounts to satisfying the discretized partial differential equation $\Delta_{\mathbf{x}} H(\mathbf{x}) = 0$ where H is the mean surface curvature. Inner fairness seeks a good distribution of parameter lines. The latter is closely related to mapping, with least distortion, a 3D triangulation to the plane [27, 28, 29]. Thus inner

fairness addresses point (c) above; but none of the criteria directly optimizes (a) or (b). Low-order *linear functionals* such as $\min \int \sum(\partial_{ij}\mathbf{x})^2$ are computationally efficient but give preference to polynomial representations of low degree. This restricts shape and can lead to unwanted flatness of the surface. While the above-mentioned geometric diffusion and mean curvature criteria aim in the right direction, such criteria are difficult to customize to allow a designer to prescribe  rather than  . One possible approach to fairing is to leverage advances in difference geometry by generating a *discrete mesh-based* representation first, for example by fairing according to [30]. The difficulty is then in switching to a parameteric surface since, typically, interpretation of the mesh points by (quasi-)interpolation leads to surface oscillations (see e.g. [31]). Similarly, switching from a level-set representation induced by a scalar field (where application of linear functionals makes sense) to the required parametric output risks loosing any of the qualities that the field was optimized for. As an alternative and key element of this survey, Section 4 therefore proposes the use of local shape hints in the form of local (guide) surface fragments.

Generating *curvature continuous surfaces* without focus on surface fairness, on the other hand, is no longer a major challenge. A number of sophisticated algorithms now exist that automatically create curvature continuous blends. Polynomial G^2 blends (where G^k indicates that adjacent patches' derivatives agree after reparameterization) include for example [32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48] for quadrilateral patches and [44,49,50,51] for triangular patches. Rational G^2 blends include [52,53] and there are non-polynomial constructions resulting in C^2 surfaces [54,55]. Yet, while local curvature continuity is helpful in analyzing surfaces, it is clearly not sufficient and certainly none of the early constructions can claim to meet both the shape and representation requirements of product-defining outer surfaces in high-end CAD/-CAM design (which need to be compatible with the industry's NURBS standard). Still, spline constructions of everywhere curvature continuous surfaces are now well enough understood to consider them a starting point rather than a goal in themselves. This state of the art sets the stage for this survey's discussion of two complementary techniques for improving the quality of curvature continuous surfaces.

3 Patch Layout

The layout of patches, i.e. the coarse parameterization of the surface, is an art requiring a good understanding of the support and approximation order of the underlying representation (Figure 2 a,b) as well as of conceptual and geometric considerations. Allowing more flexibility in the layout simplifies the layout problem and can improve the shape where a large number of patches join, such as a pole in the latitude-longitude parameterization of the sphere.

The layout typically chosen in the (untrimmed) academic setting is one induced by quadrilateral-based (tensor-product) subdivision surfaces, such as Catmull-Clark subdivision [3]; or by triangle-based (three direction box-spline) subdivision, such as Loop's [2]. Setback blending, more typical in industrial design, where primary surfaces are trimmed back and pairwise blended, can be transformed to the subdivision setting

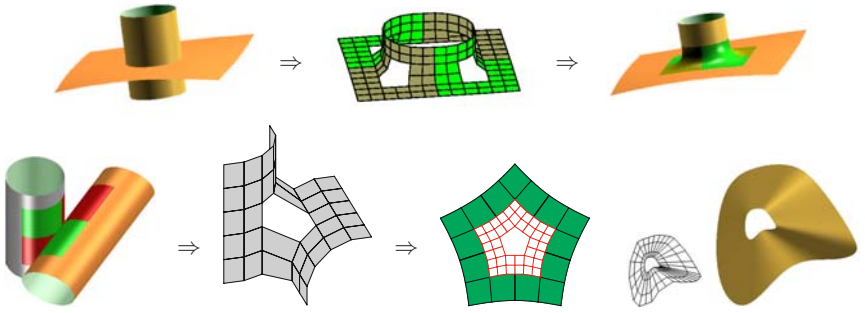


Fig. 4. Trimming and abstraction of boundary data. (*top*) The set back (trimmed) surfaces are sampled to provide a depth 2 *tensor-border* (spline representation of the boundary) with 5-sided holes. (*bottom*) Tensor-border derived from two cylinders. Such borders then have the same layout as the standard setup of subdivision (*middle*) and (*right*).

by recording the first k derivatives across the resulting boundary. We call such data along a boundary (loop) a *depth k tensor-border* (see Figure 4, *middle*).

In either case the multi-sided holes have no corresponding box-spline basis function but are filled, either with a cap of n (macro-)patches or via subdivision, i.e. by an infinite sequence of surface rings. Fundamentally, there are two types of layouts, depending on whether the edge $(t,0)$ or the edge $(1,t)$ of the domain is mapped to the predecessor's edge $(0,t)$. In the first case (Figure 5, *left* two), the layout reminds of a *sprocket*, a mechanical part with gear-teeth. Such a layout is illustrated in Figure 6, *left*. In the vicinity of the n-valent center this configuration mimics the map $z^{4/n}, z \in \mathbb{C}$ and for large n, the parameter lines are increasingly at odds with the curvature lines. Another characteristic of this layout are T-corners. A *T-corner* is the location where an edge between two distinct polynomial patches meets the midpoint of an edge of a third.

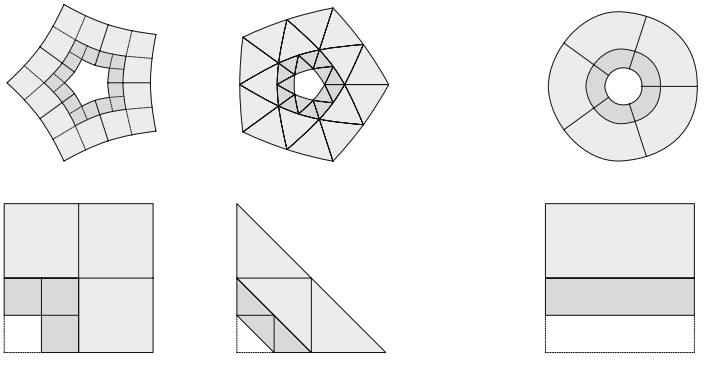


Fig. 5. Layout (*top*) Tessellating annuli generated by ct-maps (page 307) ρ and $\lambda\rho$ [44]. (*bottom*) Domains S and $S/2$ of each of the $n=5$ pieces used for the prolongation $\rho(S/2) := \lambda\rho(S)$. (*left*) Quad-sprocket (Catmull-Clark subdivision) layout; (*middle*) Triangle-sprocket (Loop subdivision) layout; (*right*) polar layout ($S/2$ is interpreted as halving only the vertical direction).

With each refinement, quad-based sprocket layout subdivision such as Catmull-Clark subdivision generates T-corners between the patches of the new surface ring and its outer ancestor. Generally, nested surface rings of quad-sprocket subdivision rings have a cascading sequence of T-corners (Figure 5, *left*). Similary, triangle-sprocket subdivision rings have a sequence of points where 5 rather than the standard 6 patches meet and one angle is π.

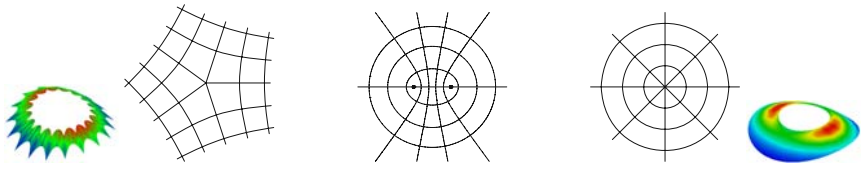


Fig. 6. (*middle* three) Iso-parameter lines of the **conformal** mappings (*left*) $z^{4/n}$; (*middle*) $\cos z$; (*right*) e^z. (Outer two) Gauss curvature images on a surface ring of (far *left*) quad- or triangle-sprocket subdivision, (far *right*) polar subdivision.

An alternative layout for subdivision is the polar layout (Figure 5, *right*) typified by the parameter lines of the map e^z (Figure 6, *right*). Here the n bounding spline segments combine to a single spline curve and the central facets are triangles that may be interpreted as quads with one collapsed edge. An intermediate configuration is the double-lens configuration according to the parameter lines of the map $\cos z = 1 + z^{4/2} + \ldots$ mediating between the two extremes [31].

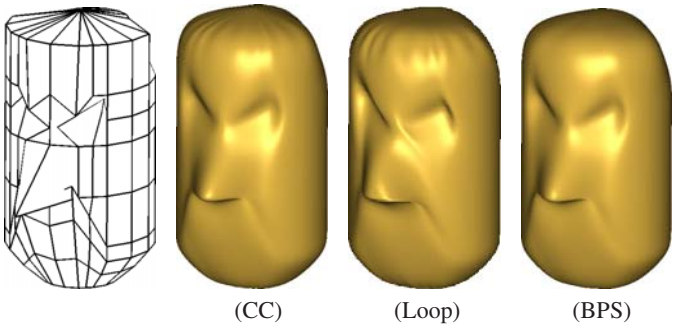


Fig. 7. Wrinkle removal by **polar layout** (CC=Catmull-Clark subdivision [3], Loop = Loop subdivision [2], BPS = Bi-3 polar subdivision [56]).

Polar layout allows for surface constructions (both by subdivision or finitely many patches [56]) that thrive on high valences (see Figure 7). Such high valences occur naturally in the design of surfaces of revolution and extruded shapes. Combining sprocket layout and polar layout [58] allows keeping the Catmull-Clark valence low by shifting high-valence connectivity to polar structures and orienting the control lines along model features (e.g. the mouth, nose and eyes in Figure 8).

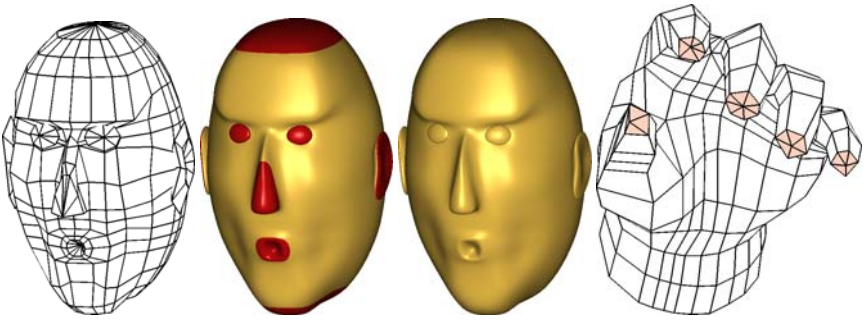


Fig. 8. Combining polar layout (for high valences, such as the centers of the eyes) with sprocket layout (for low valences, such as the 5-valent vertices surrounding the mouth or between thumb and index finger on the *right*) [57].

4 Guided Surfacing

While prior approaches attempted to address both shape design and representation in one step, guided surface construction *separates shape design from surface representation*. In the guided approach, the shape is defined via local surface fragments, called guides, that need only obey few constraints (see Figure 9, *top*,*left*). A guide typically does not match the boundary data but overlaps or leaves gaps; a guide may be of too high a degree for downstream algorithms (piecewise degree 12 is useful for some configurations) or it may have an otherwise undesirable representation that does not fit

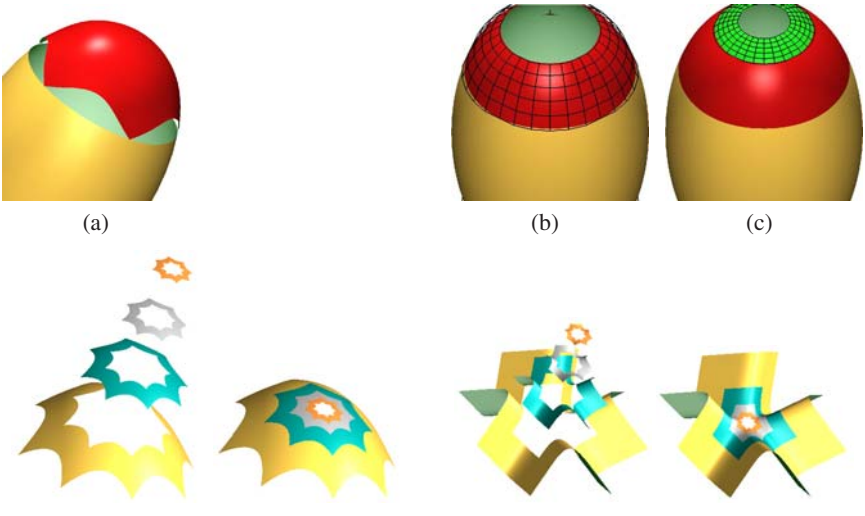


Fig. 9. Guided surface rings. (*top*) Polar layout. (a) boundary data and guide (red cap) (b) first layer, (c) second layer [43]. (*bottom*) Sprocket layout. The guided rings are shown first in an exploded view, then combined to one surface [44].

into the processing pipeline. Leveraging classical techniques of approximation theory and spline construction, the guide is replaced by a sequence of nested, smoothly joined surface rings so that the output consists of parametric pieces in standard rational or polynomial form. Guide-based algorithms can be chosen so that the final output consists of a few, of many or, in the case of subdivision, of an infinite number of patches. Both finite and subdivision constructions use guided rings to transition to the central point.

4.1 Guided Surface Rings

A *guided surface ring* is a low-degree, piecewise polynomial or rational spline approximation to a ring-shaped region of a given surface fragment, called the *guide*. The two reasons to generate guided rings are:

- to capture the shape of a complex guide surface and
- to obtain a smooth, low-degree standard (piecewise polynomial) representation.

The specific approach implemented in [44] is to generate the mth guided ring $\mathbf{x}^m$, $m = 0, 1, \ldots$ by applying an operator H that approximates (for example position and higher-order derivatives of) a guide surface $\mathbf{g}$ at parameters defined and weighted by λ^m-scaled copies of a concentric tessellation map ρ (see Figure 10):

$$\mathbf{x}^m := H(\mathbf{g} \circ \lambda^m \rho) : \quad S \times \{1, \ldots, n\} \to \mathbb{R}^3. \tag{1}$$

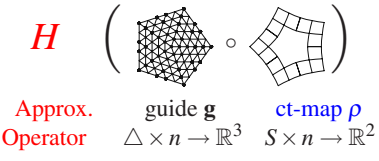


Fig. 10. Guided ring generation for sprocket layout

We explain this in more detail.

The concentric tessellation map ρ. A concentric tessellation map (short: ct-map) maps n copies of a *sector domain* S (see Figure 5, *bottom*) to an annulus in the plane:

$$\rho : S \times \{1, \ldots, n\} \to \mathbb{R}^2,$$

so that λ^m-scaled copies of this annulus join without overlap and smoothly as parameterizations to fill a disk around the origin. Characteristic maps of symmetric subdivision algorithms provide a ready source for ct-maps. For example, Figure 5, *top*, shows uniform C^2 ct-maps for three domain types S. However, the maps can also be chosen non-uniform and unsymmetric. Figure 12, *bottom* (b) shows (the extension of) such a map and illustrates its use. The ct-map relates the domain of the surface rings to that of the guide: it orients and scales higher-order derivatives that the operator H extracts from the guide surface.

The operator H. The operator H maps the composition $\mathbf{g} \circ \rho : S \times \{1,\ldots,n\} \to \mathbb{R}^3$ to a piecewise polynomial surface in $\mathbb{R}^3$. Given an intermediate patch $h(f)$ that matches the derivatives of a given map f defined on S, H prolongs it so that consecutive rings $\mathbf{x}^m, \mathbf{x}^{m+1}$ join smoothly (Figure 9) and the outermost ring, $\mathbf{x}^0$, smoothly joins the surrounding multi-sided boundary data, the tensor-border. In typical guided surface constructions, *higher-order jets* (collections of derivatives) of the composition $f := \mathbf{g} \circ \rho$ are well-defined because the rays that form the segment boundaries of the ct-map ρ match the domain boundaries of the polynomial pieces of the guide $\mathbf{g}$. Then, for example the operator $h := h^{55}$ for tensor-product patches, generates a patch of degree $(5,5)$ as follows. For each corner, h^{55} samples the second order expansion of f, and converts it to a 3×3 group of Bernstein-Bézier coefficients.

$$\begin{matrix} \partial_t^2 f & \partial_s \partial_t^2 f & \partial_s^2 \partial_t^2 f \\ \partial_t f & \partial_s \partial_t f & \partial_s^2 \partial_t f \\ f & \partial_s f & \partial_s^2 f \end{matrix} \quad \Rightarrow \quad \begin{matrix} \circ & \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ & \circ \\ \cdot & \cdot & \cdot & \circ & \circ & \circ \\ \cdot & \cdot & \cdot & \circ & \circ & \circ \\ \cdot & \cdot & \cdot & \circ & \circ & \circ \end{matrix} \tag{2}$$

By combining the four groups of nine coefficients per corner, h^{55} defines the 6×6 coefficients of a polynomial patch of degree bi-5. Alternatively, h can be an operator h^{33} that substitutes, for each bi-5 patch, a 3×3 array of bi-3 patches, i.e. a C^2 spline in B-spline form with single knots in the interior and four-fold knots at the boundary [47]. Clearly many other choices for h are possible.

Guided patchworks. Let ρ be a ct-map and let the guide $\mathbf{g}$ be defined as

$$\mathbf{g} : \mathbb{R}^2 \to \mathbb{R}^3, \quad (u,v) \mapsto (x,y,z). \tag{3}$$

The contraction of the smoothly connected annuli $\{\lambda^m \rho\}_m$ is inherited by the sequence of compositions $\{\mathbf{g} \circ \lambda^m \rho\}_m$ and the sampled C^2 guided rings $\mathbf{x}^m := H(\mathbf{g} \circ \lambda^m \rho)$ will join to form a C^2 surface in $\mathbb{R}^3$ [44, Lemma 4] (as illustrated in Figure 9). There are a number of possible combinations of h (and hence H) and ρ. For example, guided C^2 quad-sprocket patchworks in [44] use as ρ the characteristic ring of Catmull-Clark subdivision (Figure 5 *top,left*) and use $h := h^{66}$, which samples up to 3rd order and averages coefficients determined by multiple corners. The operator h samples $\mathbf{g} \circ \lambda^m \rho$ at the corners of the L-shaped segment of a quad-sprocket construction. – For a second example, guided triangle-sprocket patchworks use the characteristic ring of Loop subdivision as ρ and apply an operator h^8 [44] that returns patches of total degree 8. The finite patchworks defined by these combinations of ρ and H (and a similar one for polar layout) can reproduce quadratic expansions at the central point. This yields a systematic way to generate C^2 subdivision surfaces, the topic of the next section.

4.2 Guided Subdivision

A sequence of nested surface rings yields a surface ring structure just like that of standard subdivision (see Figure 11). While standard mesh-based subdivision generates

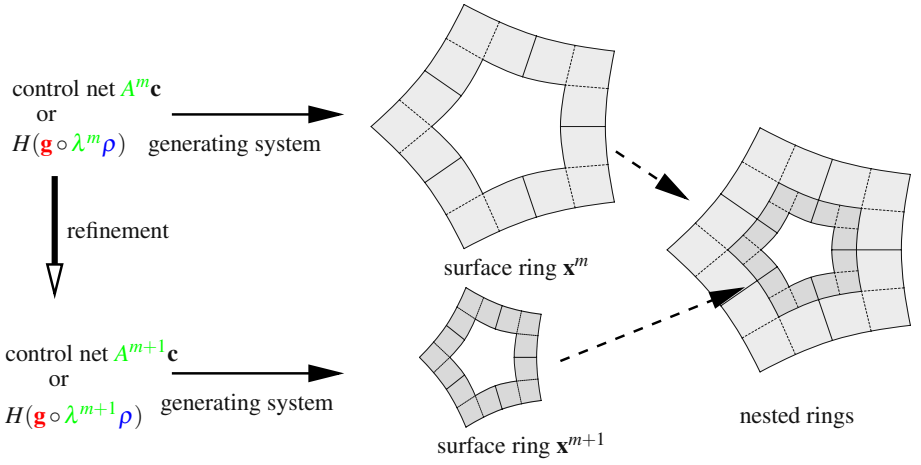


Fig. 11. Standard **generalized subdivision and guided subdivision** both generate a sequence of nested surface rings: Generalized subdivision surface rings are guided by a subdivision matrix A, guided subdivision surface rings are guided by an explicit map $\mathbf{g}$ [47].

nested surface rings by applying powers of a square subdivision matrix A to mesh nodes (control points) $\mathbf{c}$ and interpreting the resulting nodes as, say B-spline control points, guided subdivision composes scaled copies of the ct-map with the guide and then re-approximates to obtain nested surface rings. Indeed the finite union of guided rings is easily constructed to be a C^2 surface. And if the degree of the surface rings is chosen to be bi-6, the guided rings reproduce the quadratic expansion of the C^2 guide at the central point so that the infinite union is C^2 [44, Thm 1]. There are also constructions of degree as low as bi-3 that preserve the shape so that the difference in quality is only visible by comparing Gauss curvature. Figure 10 of [47] shows the correlation of lowering the degree with a deterioration of the curvature distribution (as does the analogous Figure 14 of [48] for finite polynomial constructions).

If the guide is (piecewise) polynomial then the construction of the contracting guided patchwork rings is stationary and guided subdivision has a fast evaluation algorithm based on superposition of eigenfunctions. A specific construction of a C^2 subdivision algorithm without explicit use of the guide is given in [43]. This algorithm maps jets to jets rather than control points in $\mathbb{R}^3$ to points in $\mathbb{R}^3$, of a refined mesh. A general theory of C^k subdivision constructions analogous to guided subdivision is laid out in [59, Ch 7]. Additional images can be found in [60].

4.3 Construction with Finitely Many Patches

Complex surface blends, for example when capping a C^2 spline surface by n patches, require an increase in either the degree or the number of pieces compared to the surrounding regular spline surface. Typically, the new degrees of freedom do not match the formal continuity constraints and this results in an under-constrained problem when

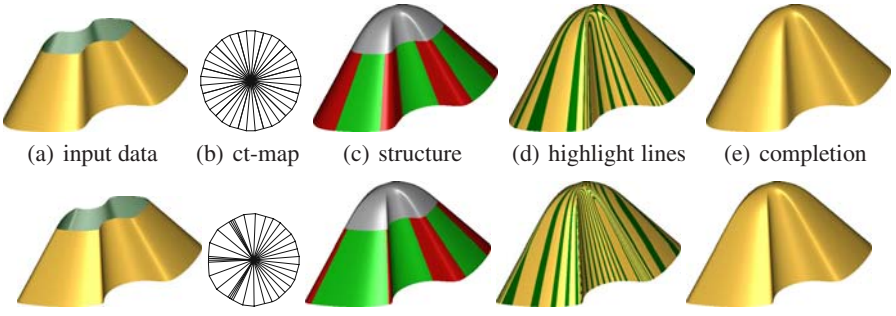


Fig. 12. Completion of a piecewise cone. Cone pieces (light green) in (c) and pairwise blends (dark red) by n patches of degree (6,5) (gray) using (*top*) a uniform and (*bottom*) a non-uniform ct-map [61]. The non-uniform ct-map allows us to match the sharper input features on the left lobe.

fitting with finitely many patches. Guided surfacing effectively localizes the technical challenges while preserving global shape, even when the number of patches, their layout and/or their degree are high [45, 46, 48]. Typically, the finite constructions benefit from a few guided rings to transition from the boundary data to the central cap. The guide stabilizes the transition, so that we do not observe the fluctuations of Figure 2 (f). Often, the rings and the final cap can be combined into macro-patches or even splines for implementation. The spline-based approach trades degree for number of pieces. We recently derived an algorithm for generating C^2 surfaces with sprocket layout consisting of n splines of degree bi-5 and a finite polar C^2 construction of degree (6,5) [61] (see Figure 12); constructions of even lower-degree are possible for simpler tensor-borders.

4.4 Fitting a Guide

If the design(er) does not provide the guide surface, and/or specifies partial information such as the position and normal of the central point, the following default construction of a piecewise polynomial C^2 guide for sprocket layout can be used [51]. We construct a C^2 map $\mathbf{g}$ consisting of n polynomial pieces of total degree 5 (see Figure 13) in BB-form:

$$\mathbf{g}^\ell(u,v) := \sum_{i+j+k=5} \mathbf{g}^\ell_{ijk} \frac{5!}{i!j!k!} u^i v^j (1-u-v)^k, \quad \ell = 1,\ldots,n.$$

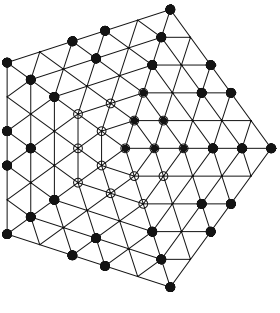


Fig. 13. Guide construction. The free control points $\mathbf{g}^\ell_{ijk}$ after enforcing C^2 constraints between the polynomial pieces of degree 5 are marked as $\bullet$.

While there is no restriction on the degree of the guide surface since it is re-sampled and hence does not influence the degree of the output. However, in our experience guide surfaces consisting of n C^2-connected triangular patches of total degree 5 suffice to approximate well second order boundary data (tensor-borders of depth 2) and there are no

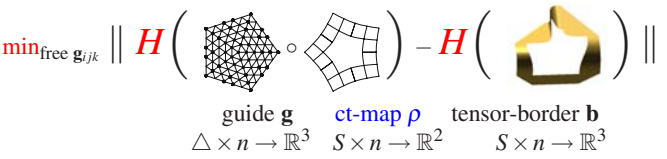


Fig. 14. Derivation of a **default guide** surface for sprocket layout. H – Approximation operator (page 308)

noticeable improvements in the final surface if the degree is higher. Conversely guides based on single polynomials or lower-degree piecewise polynomial typically fail to capture the existing boundary data.

With this Ansatz, enforcing the C^1 and C^2 constraints in terms of the BB-coefficients leaves as free coefficients (shown as black bullets in Figure 13)

$$\mathbf{g}^*_{ijk,i+j+k=5}: \quad \mathbf{g}^0_{ijk}, j+k \le 2, \quad \mathbf{g}^\ell_{ij0}, j = 3,4,5, \quad \mathbf{g}^\ell_{122}, \mathbf{g}^\ell_{023}, \mathbf{g}^\ell_{032}.$$

That is, the center quadratic polynomial is to be determined and each sector has some extra degrees of freedom. For elliptic shapes, either the user should provide the location of the central point or we set it as the limit point of a subdivision scheme. To set the remaining free coefficients, we minimize the deviation of the guide from the boundary data in the sense of Figure 14: given the ct-map ρ and the tensor-border $\mathbf{b}$, we minimize

$$\min_{\text{free}\mathbf{g}^*_{ijk}} \|H(\mathbf{g}\circ\rho) - H(\mathbf{b})\|_2^2. \tag{4}$$

Guide creation and re-approximation can be alternated by using the preceding surface ring as boundary data $\mathbf{b}$.

5 A C^2 Accelerated Bi-3 Guided Subdivision

To illustrate the level of flexibility provided by guided surfacing, we consider a surface ring construction that samples with increasing density [62]. We first consider the sprocket (Catmull-Clark) layout Figure 15, *left*. At level m, each of the three quads of an L-shaped sector is partitioned into $2^m \times 2^m$ subquads. The operator h^{33} (see page 308) is applied on each subquad, creating, after removal of the internal knots, a C^2 bi-3 spline that joins C^2 with its neighbor spline in the surface ring. The construction approximates the piecewise polynomial $\mathbf{g}\circ\rho : [0..1]^2 \to \mathbb{R}$ by h^{33} up to second order at the corners of the 34^m subquads of each quad-sprocket segment $\mathbf{x}^m_\ell$, For polar layout, there are 2^m subquads (see Figure 15, *right*) and the construction approximates the piecewise polynomial up to second order at the corners of the 2^m subquads of each polar segment $\mathbf{x}^m_\ell$. We call such schemes *accelerated*. To certify that the resulting subdivision surfaces are generically curvature continuous, [62, Thm 6] tracked a sequence of local quadratic functions and showed their convergence to the quadratic Taylor expansion of $\mathbf{g}$ at the central point.

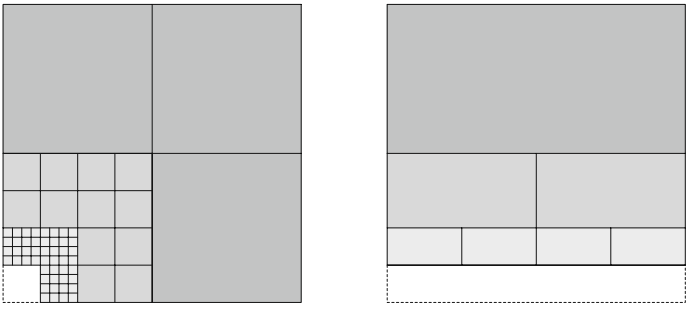


Fig. 15. Structure of a sector of **accelerated C^2 bi-3 subdivision**. (*left*) Quad-sprocket (Catmull-Clark) layout and (*right*) polar layout [62].

Acceleration therefore circumvents one of the key assumptions that lead to the lower bounds on the degree of curvature continuous subdivision surfaces derived in [63, 64]. Acceleration allows us to build C^2 surfaces consisting of (infinitely many) polynomial pieces of degree (3,3), the continuity-and-degree combination hoped for since 1978 [3]. While the construction for the sprocket configurations seems hardly practical, acceleration for polar configurations is natural and has been turned into a practical C^2 bi-3 subdivision algorithm [65].

Acknowledgements

The images were generated with the surface analysis tool *BezierView*. This work was supported by the National Science Foundation Grant 0728797.

References

1. Várady, T., Rockwood, A.: Geometric construction for setback vertex blending. Computer-aided Design 29(6), 413–425 (1997)
2. Loop, C.: Smooth Subdivision Surfaces Based on Triangles. PhD thesis, Mathematics, University of Utah (1987)
3. Catmull, E., Clark, J.: Recursively generated B-spline surfaces on arbitrary topological meshes. Computer-Aided Design 10, 350–355 (1978)
4. Peters, J., Reif, U.: Shape characterization of subdivision surfaces – basic principles. Computer-Aided Geometric Design 21(6), 585–599 (2004)
5. Karciauskas, K., Peters, J., Reif, U.: Shape characterization of subdivision surfaces – case studies. Computer-Aided Geometric Design 21(6), 601–614 (2004)
6. Zorin, D., Schröder, P.: A unified framework for primal/dual quadrilateral subdivision schemes. Comput. Aided Geom. Design 18(5), 429–454 (2001); Subdivision algorithms (Schloss Dagstuhl, 2000)
7. Bloor, M., Wilson, M.J.: Using partial differential equations to generate free form surfaces. Computer Aided Design 22(4), 202–212 (1990)
8. Greiner, G.: Variational design and fairing of spline surfaces. Computer Graphics Forum 13(3), C/143–C/154 (1994)

9. Hubeli, A., Gross, M.: Fairing of non-manifolds for visualization. In: Proceedings Visualization 2000, IEEE Computer Society Technical Committe on Computer Graphics, pp. 407–414 (2000)
10. Moreton, H.P., Séquin, C.H.: Functional optimization for fair surface design. Computer Graphics 26(2), 167–176 (1992)
11. Sapidis, N.: Designing Fair Curves and Surfaces. SIAM, Philadelphia (1994)
12. Welch, W., Witkin, A.: Variational surface modeling. Computer Graphics (SIGGRAPH 1992 Proceedings) 26(2), 157–166 (1992)
13. Kobbelt, L., Campagna, S., Vorsatz, J., Seidel, H.P.: Interactive multi-resolution modeling on arbitrary meshes. In: Cohen, M. (ed.) SIGGRAPH 1998 Conference Proceedings. Annual Conference Series, ACM SIGGRAPH, pp. 105–114. Addison Wesley, Reading (1998)
14. Kobbelt, L., Hesse, T., Prautzsch, H., Schweizerhof, K.: Interactive mesh generation for FE-computation on free form surfaces. Engng. Comput. 14, 806–820 (1997)
15. Welch, W., Witkin, A.: Free–Form shape design using triangulated surfaces. In: Glassner, A. (ed.) Proceedings of SIGGRAPH 1994, Orlando, Florida, July 24-29. Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, pp. 247–256. ACM Press, New York (1994)
16. do Carmo, M.P.: Riemannian Geometry. Birkhäuser Verlag, Boston (1992)
17. Taubin, G.: A signal processing approach to fair surface design. In: Cook, R. (ed.) SIGGRAPH 1995 Conference Proceedings, Los Angeles, California, August 6-11. Annual Conference Series, ACM SIGGRAPH, pp. 351–358. Addison Wesley, Reading (1995)
18. Kobbelt, L.: Discrete fairing. In: Goodman, T., Martin, R. (eds.) Proceedings of the 7th IMA Conference on the Mathematics of Surfaces (IMA 1996), Winchester, UK, September 1997. Mathematics of Surfaces, vol. VII, pp. 101–130. Information Geometers (1997)
19. Desbrun, M., Meyer, M., Schröder, P., Barr, A.H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: Rockwood, A. (ed.) Siggraph 1999, Los Angeles. Annual Conference Series, ACM Siggraph, pp. 317–324. Addison Wesley Longman, Amsterdam (1999)
20. Meyer, M., Desbrun, M., Schroder, P., Barr, A.: Discrete differential-geometry operators for triangulated 2-manifolds. Visualization and Mathematics 3, 34–57 (2002)
21. Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugates. Experimental Mathematics 2(1), 15–36 (1993)
22. Westermann, R., Johnson, C., Ertl, T.: A level-set method for flow visualization. In: Ertl, T., Hamann, B., Varshney, A. (eds.) Proceedings Visualization 2000, IEEE Computer Society Technical Committee on Computer Graphics, pp. 147–154 (2000)
23. Jin, M., Kim, J., Luo, F., Lee, S., Gu, X.: Conformal surface parameterization using Euclidean Ricci flow. Technical Report, http://www.cise.ufl.edu/~gu/publications/technical_report.htm
24. Kimmel, R.: Intrinsic scale space for images on surfaces: The geodesic curvature flow. Graphical models and image processing: GMIP 59(5), 365–372 (1997)
25. Malladi, R., Sethian, J.A.: Flows under min/max curvature flow and mean curvature: applications in image processing. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1064, p. 251. Springer, Heidelberg (1996)
26. Schneider, R., Kobbelt, L.: Generating fair meshes with G^1 boundary conditions. In: Martin, R., Wang, W. (eds.) Proceedings of the Conference on Geometric Modeling and Processing (GMP 2000), April 10-12, pp. 251–261. IEEE, Los Alamitos (2000)
27. Floater, M.S.: Parametrization and smooth approximation of surface triangulations. Computer Aided Geometric Design 14(3), 231–250 (1997)
28. Sheffer, A., de Sturler, E.: Surface parameterization for meshing by triangulation flattening. In: Proc. 9th International Meshing Roundtable, pp. 161–172 (2000)

29. Gu, X., Gortler, S.J., Hoppe, H.: Geometry images. In: Hughes, J. (ed.) SIGGRAPH 2002 Conference Proceedings. Annual Conference Series, pp. 335–361. ACM Press/ACM SIGGRAPH (2002)
30. Hildebrandt, K., Polthier, K.: Constraint-based fairing of surface meshes. In: Belyaev, A., Garland, M. (eds.) SGP 2007: Eurographics Symposium on Geometry Processing, pp. 203–212. Eurographics Association (2007)
31. Karčiauskas, K., Peters, J.: Lens-shaped surfaces and C^2 subdivision. Computing (to appear, 2009)
32. Hahn, J.: Filling polygonal holes with rectangular patches. In: Theory and practice of geometric modeling, Blaubeuren, 1988, pp. 81–91. Springer, Berlin (1989)
33. Gregory, J.A., Hahn, J.M.: A C^2 polygonal surface patch. Comput. Aided Geom. Design 6(1), 69–75 (1989)
34. Ye, X.: Curvature continuous interpolation of curve meshes. Computer Aided Geometric Design 14(2), 169–190 (1997)
35. Reif, U.: TURBS—topologically unrestricted rational B-splines. Constructive Approximation 14(1), 57–77 (1998)
36. Reif, U.: Analyse und Konstruktion von Subdivisionsalgorithmen für Freiformflächen beliebiger Topologie. Shaker Verlag, Aachen (1999)
37. Prautzsch, H.: Freeform splines. Computer Aided Geometric Design 14(3), 201–206 (1997)
38. Peters, J.: Curvature continuous spline surfaces over irregular meshes. Computer-Aided Geometric Design 13(2), 101–131 (1996)
39. Gregory, J.A., Zhou, J.: Irregular C^2 surface construction using bi-polynomial rectangular patches. Comput. Aided Geom. Design 16(5), 423–435 (1999)
40. Loop, C.: Second order smoothness over extraordinary vertices. In: Symp. Geom. Processing, pp. 169–178 (2004)
41. Loop, C.T., Schaefer, S.: G ^{2} tensor product splines over extraordinary vertices. Comput. Graph. Forum 27(5), 1373–1382 (2008)
42. Peters, J.: C^2 free-form surfaces of degree (3,5). Computer-Aided Geometric Design 19(2), 113–126 (2002)
43. Karčiauskas, K., Myles, A., Peters, J.: A C^2 polar jet subdivision. In: Scheffer, A., Polthier, K. (eds.) Proceedings of Symposium of Graphics Processing (SGP), Cagliari, Italy, June 26-28, pp. 173–180. ACM Press, New York (2006)
44. Karčiauskas, K., Peters, J.: Concentric tesselation maps and curvature continuous guided surfaces. Computer-Aided Geometric Design 24(2), 99–111 (2007)
45. Karčiauskas, K., Peters, J.: Parameterization transition for guided C^2 surfaces of low degree. In: Sixth AFA Conference on Curves and Surfaces Avignon, June 29-July 5, 2006, pp. 183–192 (April 2007)
46. Karčiauskas, K., Peters, J.: Guided C^2 spline surfaces with V-shaped tessellation. In: Winkler, J., Martin, R. (eds.) Mathematics of Surfaces, pp. 233–244 (2007)
47. Karčiauskas, K., Peters, J.: On the curvature of guided surfaces. Computer Aided Geometric Design 25(2), 69–79 (2008)
48. Karčiauskas, K., Peters, J.: Guided spline surfaces. Computer Aided Geometric Design, 1–20 (2009 N1)
49. Prautzsch, H., Umlauf, G.: Triangular G^2splines. In: Laurent, P.J., A. LeMéhauté, L.S. (eds.) Curve and Surface Design, pp. 335–342. Vanderbilt University Press (2000)
50. Bohl, H., Reif, U.: Degenerate Bézier patches with continuous curvature. Comput. Aided Geom. Design 14(8), 749–761 (1997)
51. Karčiauskas, K., Peters, J.: Assembling curvature continuous surfaces from triangular patches. SMI 26/105, Computers and Graphics (2009), http://dx.doi.org/10.1016/j.cag.2009.03.015

52. Navau, J.C., Garcia, N.P.: Modeling surfaces from meshes of arbitrary topology. Comput. Aided Geom. Design 17(7), 643–671 (2000)
53. Grimm, C.M., Hughes, J.F.: Modeling surfaces of arbitrary topology using manifolds. Computer Graphics 29, 359–368 (1995); (Annual Conference Series)
54. Ying, L., Zorin, D.: A simple manifold-based construction of surfaces of arbitrary smoothness. ACM TOG 23(3), 271–275 (2004)
55. Levin, A.: Modified subdivision surfaces with continuous curvature. ACM Transactions on Graphics 25(3), 1035–1040 (2006)
56. Karčiauskas, K., Peters, J.: Bicubic polar subdivision. ACM Trans. Graph. 26(4), 14 (2007)
57. Myles, A., Karčiauskas, K., Peters, J.: Extending Catmull-Clark subdivision and PCCM with polar structures. In: PG 2007: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications, Washington, DC, USA, pp. 313–320. IEEE Computer Society, Los Alamitos (2007)
58. Myles, A., Karčiauskas, K., Peters, J.: Pairs of bi-cubic surface constructions supporting polar connectivity. Comput. Aided Geom. Des. 25(8), 621–630 (2008)
59. Peters, J., Reif, U.: Subdivision Surfaces. Geometry and Computing, vol. 3. Springer, New York (2008)
60. SurfLab, http://www.cise.ufl.edu/research/surflab
61. Karčiauskas, K., Peters, J.: Finite curvature continuous polar patches. In: Hancock, E., Martin, R. (eds.) IMA Mathematics of Surfaces XIII Conference (in press, 2009)
62. Karčiauskas, K., Peters, J.: Guided subdivision. Technical Report 2008-464, Dept CISE, University of Florida (2008), posted since 2005 at:
http://www.cise.ufl.edu/research/SurfLab/papers.shtml
63. Reif, U.: A degree estimate for subdivision surfaces of higher regularity. Proc. Amer. Math. Soc. 124(7), 2167–2174 (1996)
64. Prautzsch, H.: Smoothness of subdivision surfaces at extraordinary points. Adv. Comput. Math. 9(3-4), 377–389 (1998)
65. Myles, A.: Curvature-continuous Bicubic Subdivision Surfaces for Polar Configurations. PhD thesis, Dept. CISE, U Florida, USA (2008)

An Iterative Algorithm with Joint Sparsity Constraints for Magnetic Tomography

Francesca Pitolli[1] and Gabriella Bretti[1,2]

[1] Università di Roma "La Sapienza", Dip. Me.Mo.Mat.
Via A. Scarpa 16, 00161 Roma, Italy
pitolli@dmmm.uniroma1.it
[2] Università Campus Bio-medico
Via Àlvaro del Portillo, 21 - 00128 Roma, Italy

Abstract. Magnetic tomography is an ill-posed and ill-conditioned inverse problem since, in general, the solution is non-unique and the measured magnetic field is affected by high noise.

We use a joint sparsity constraint to regularize the magnetic inverse problem. This leads to a minimization problem whose solution can be approximated by an iterative thresholded Landweber algorithm. The algorithm is proved to be convergent and an error estimate is also given.

Numerical tests on a bidimensional problem show that our algorithm outperforms Tikhonov regularization when the measurements are distorted by high noise.

1 Introduction

The magnetic tomography aims at spatially resolving an unknown vector-valued current distribution from its magnetic field measured in the outer space.

Magnetic tomography has applications in many fields, i.e. geophysics, archaeological investigations, medicine, microelectronic, nondestructive testing. In particular, we focus our interest on magnetoencephalography (MEG) and nondestructive evaluation (NDE) of structures. MEG aims to identify the regions of the brain where the neural activity is located by detecting the regions where bioelectric currents flow, while NDE aims to localize cracks and corrosion damage which appear as perturbations of the current distribution within a given material [19]. In both cases an image of the current distribution within the object under investigation has to be reconstructed in a noninvasive way, i.e. by the magnetic field measured by sensors located in the outer space.

Since the magnetic field decreases very fast as the distance between the current source and the sensor position increases, the measured magnetic field can be very weak and affected by high noise. Usually, the measurements are performed by very sophisticated instrumentations based on Superconducting QUantum Interference Device (SQUID) magnetometers that can measure magnetic fields generated by deep currents [19]. Even if SQUID magnetometers are able to measure very weak field, nevertheless the measurements are distorted by high noise and a sophisticated data analysis is needed.

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 316–328, 2010.

In magnetic tomography, the measurements do not give an immediate current image. In order to produce a current image, first of all, we need a model to relate the current distribution and the external magnetic field; then, we have to solve an inverse problem. In literature, this problem is known as magnetic inverse problem, and, in general, do not have a unique solution, so that regularization techniques are needed [2], [14], [17].

The well-known Tikhonov regularization, which uses a quadratic constraint, gives good results when the quantities under observation are equally distributed in time or space [1], [8]. Actually, in MEG and NDE applications the regions where the currents flow are usually small. This means that the current distribution we want to reconstruct is spatially inhomogeneous and can be represented as a sum of weighted basic currents where only few terms are relevant. To reconstruct quantities with sparse patterns a regularization technique based on *sparsity constraints* can be used; this leads to a thresholded Landweber algorithm which is able to approximate the solution of a linear inverse problem in an efficient way [5].

Our aim is to solve the magnetic inverse problem by using a joint sparsity constraint as a regularization technique. Joint sparsity has been introduced in [11] in order to deal with problems where the quantities to reconstruct are vector-valued functions. The solution of the MEG inverse problem with joint sparsity constraints has been addressed in [10]. Here, we focus our attention on NDE applications of the magnetic tomography. The forward and inverse magnetic NDE problems are addressed in Section 2. Then, in Section 3 we introduce a joint sparsity constraint as a regularization term. An efficient algorithm to solve the problem is given in Section 4. Finally, some numerical tests on a bidimensional problem are shown in Section 5.

2 The Forward and Inverse Magnetic NDE Problems

NDE problems can be modeled by the Maxwell's equations for a polarizable and magnetizable macroscopic media [2]. In particular, the electric field $\mathbf{E}$ and the magnetic field $\mathbf{B}$ can be described by the quasi-static Maxwell's equations

$$\begin{aligned} \operatorname{curl} \mathbf{E} &= 0 \ , \\ \operatorname{curl} \mathbf{B} &= \mu_0 \mathbf{J} \ , \\ \operatorname{div} \mathbf{B} &= 0 \ , \end{aligned} \tag{1}$$

where $\mathbf{J}$ is the current density in the medium and μ_0 is the magnetic permeability of the vacuum.

Due to the conservation of the charge, the current density satisfies

$$\operatorname{div} \mathbf{J} = 0 \ . \tag{2}$$

In the following we assume that the volume V_0 where the current flows is a slab of thickness d, with the surfaces of the slab at $z = \pm d/2$. The magnetometers are located on an horizontal plane $\mathbf{\Pi}$ at $z = \pi$, with

$$\pi := \operatorname{dist}(\partial V_0, \mathbf{\Pi}) > \frac{d}{2} \ , \tag{3}$$

and they measure only B_z, the component of the magnetic field along $\mathbf{e}_z$, the normal w.r.t. to $\mathbf{\Pi}$ (see Fig. 1).

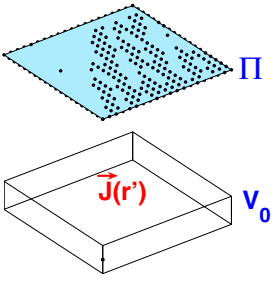


Fig. 1. The NDE problem setting

From the quasi-static Maxwell's equations it follows that the magnetic field $\mathbf{B}$ generated by the current $\mathbf{J}$ obeys the *Biot-Savart law* [2], [13], [17]

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \int_{V_0} \mathbf{J}(\mathbf{r}\,') \times \frac{\mathbf{r} - \mathbf{r}\,'}{|\mathbf{r} - \mathbf{r}\,'|^3}\, d\mathbf{r}\,' \ . \tag{4}$$

Here, $\mathbf{v} \times \mathbf{w}$ is the usual cross product of vectors in $\mathbb{R}^3$ and $|\mathbf{v}|$ is the Euclidean norm of the vector $\mathbf{v}$.

For $B_z(\mathbf{r})$ we have

$$\begin{aligned} B_z(\mathbf{r}) &= \frac{\mu_0}{4\pi} \int_{V_0} \frac{\mathbf{J}(\mathbf{r}\,') \times (\mathbf{r} - \mathbf{r}\,')}{|\mathbf{r} - \mathbf{r}\,'|^3} \cdot \mathbf{e}_z(\mathbf{r})\, d\mathbf{r}\,' = \\ &= \frac{\mu_0}{4\pi} \int_{V_0} \frac{\mathbf{e}_z(\mathbf{r}) \times (\mathbf{r}\,' - \mathbf{r})}{|\mathbf{r}\,' - \mathbf{r}|^3} \cdot \mathbf{J}(\mathbf{r}\,')\, d\mathbf{r}\,' \ , \ \mathbf{r} \in \mathbf{\Pi} \ , \end{aligned} \tag{5}$$

where the equality in the second line follows from the relation $\mathbf{v} \times \mathbf{w} \cdot \mathbf{z} = -\mathbf{z} \times \mathbf{w} \cdot \mathbf{v}$, holding for any $\mathbf{v}, \mathbf{w}, \mathbf{z} \in \mathbb{R}^3$.

Now, let $\mathbf{q}_l \in \mathbf{\Pi}$, $l = 1, \ldots, N$, be the locations of the magnetometers and let

$$\mathcal{B}(\mathbf{J}, \mathbf{q}_l) := B_z(\mathbf{q}_l) = \frac{\mu_0}{4\pi} \int_{V_0} \frac{\mathbf{e}_z(\mathbf{q}_l) \times (\mathbf{r}\,' - \mathbf{q}_l)}{|\mathbf{r}\,' - \mathbf{q}_l|^3} \cdot \mathbf{J}(\mathbf{r}\,')\, d\mathbf{r}\,' \tag{6}$$

be the normal component of the magnetic field evaluated in $\mathbf{q}_l$.

Observe that the magnetic field at different values of z are related by inward continuation, so that we cannot obtain additional information by measuring the magnetic field on planes at different heights. By the way, since $\operatorname{curl} \mathbf{B} = 0$ and $\operatorname{div} \mathbf{B} = 0$ in the outer space, information on the z-component of the magnetic field is sufficient to determine the whole magnetic field [15].

The magnetic NDE inverse problem aims at reconstructing the current distribution $\mathbf{J}$ starting from the normal component of the magnetic field, measured in $\mathbf{q}_l$, $l = 1, \ldots, N$. In order to identify the current sources from the measurements $\mathbf{M} = \{m_1, \ldots, m_N\}$ we have to minimize the discrepancy

$$\Delta(\mathbf{J}) := \|\mathbf{G}(\mathbf{J}) - \mathbf{M}\|^2_{\mathbb{R}^N} \ , \tag{7}$$

where

$$\mathbf{G}(\mathbf{J}) = \{\mathcal{B}(\mathbf{J}, \mathbf{q}_1), ..., \mathcal{B}(\mathbf{J}, \mathbf{q}_N)\} \ . \tag{8}$$

Unfortunately, this is a strongly ill-posed problem since there exist silent currents that do not produce magnetic field in the outer space, so that non unique solutions can be expected. Moreover, the intensity of the magnetic field decreases fast as the height of the plane $\mathbf{\Pi}$ increases, so that the measured magnetic field can be very low and affected by high noise. For these reasons the minimization of the discrepancy might not be feasible and some regularization technique is required. This will be the subject of the following section.

3 The Magnetic NDE Inverse Problem with Sparsity Constraints

We assume that the current distribution we want to reconstruct is spatially inhomogeneous and can be represented as a sum of weighted basic currents belonging to a dictionary, such that only few terms in the sum are relevant. More formally, we assume that [6].

Definition 1. *The current density* $\mathbf{J} = (J_1, J_2, J_3) \in L_2(V_0; \mathbb{R}^3)$ *is* sparsely *represented by a suitable dictionary* $\mathcal{D} := (\psi_\lambda)_{\lambda \in \mathbf{\Lambda}}$, *i.e.*

$$J_\ell \approx \sum_{\lambda \in \mathbf{\Lambda}_S} j_\lambda^\ell \, \psi_\lambda \ , \qquad j_\lambda^\ell = \langle J_\ell, \psi_\lambda \rangle \ , \qquad \ell = 1, 2, 3 \ , \tag{9}$$

where $\mathbf{\Lambda}_S \subset \mathbf{\Lambda}$ *is the set of the* few *significant coefficients* j_λ^ℓ.

As a dictionary we choose a stable multiscale basis $(\psi_\lambda)_{\lambda \in \mathbf{\Lambda}} \in L_2(V_0, \mathbb{R})$, i.e. a function basis satisfying the inequality

$$C_1 \|f\|^2_{L_2(V_0)} \leq \sum_{\lambda \in \mathbf{\Lambda}} |\langle f, \psi_\lambda \rangle|^2 \leq C_2 \|f\|^2_{L_2(V_0)} \ , \quad \lambda \in \mathbf{\Lambda} \ , \tag{10}$$

for all $f \in L_2(V_0)$. Suitable dictionaries are, for instance, wavelets [4] or frames [3].

In order to enforce sparsity on the solution of the magnetic NDE inverse problem we use a sparsity constraint as follows.

Given a set of magnetic field measurements $\mathbf{M}$, *determine a configuration of the current density* $\mathbf{J}$ *that minimizes the functional*

$$\mathcal{J}_{\mathbf{\Psi}}(\mathbf{J}) := \mathbf{\Delta}(\mathbf{J}) + \mathbf{\Psi}_{\mathcal{D}}(\mathbf{J}) \ , \tag{11}$$

where $\mathcal{D}$ *is a dictionary chosen in order the solution of the minimum problem is sparsely represented, and* $\mathbf{\Psi}_{\mathcal{D}}$ *is a suitable* sparsity measure *w.r.t.* $\mathcal{D}$.

A sparsity measure for the regularization of linear ill-posed problems when the quantity to reconstruct is a scalar function f, has been introduced in [5]. In this case

$$\mathbf{\Psi}_{\mathcal{D}}(f) := \sum_{\lambda \in \mathbf{\Lambda}} v_\lambda \, |\langle f, \psi_\lambda \rangle|^p \ , \qquad p \geq 1 \ , \tag{12}$$

where $(v_\lambda)_{\lambda \in \Lambda}$ are nonnegative weights. The solution of the related minimization problem leads to an iterative thresholded Landweber algorithm.

Since in our problem the current $\mathbf{J}$ is a vector-valued function, a suitable sparsity measure is the joint sparsity introduced in [11].

Definition 2. *Let $\mathbf{F} = (F_1, F_2, F_3)$ be a vector-valued function in $L_2(V_0, \mathbb{R}^3)$, and let $(\psi_\lambda)_{\lambda\in\mathbf{\Lambda}}$ be a dictionary. Let us denote by*

$$\mathbf{f}_\lambda = (f_\lambda^1, f_\lambda^2, f_\lambda^3) \quad \textit{with} \quad f_\lambda^\ell = \langle F_\ell, \psi_\lambda \rangle \tag{13}$$

the decomposition coefficients of $\mathbf{F}$ and by $\|\mathbf{f}_\lambda\|_p$ the usual p-norm for vectors.

Given the positive sequences $(\theta_\lambda)_{\lambda\in\mathbf{\Lambda}}$, $(\rho_\lambda)_{\lambda\in\mathbf{\Lambda}}$, $(\omega_\lambda)_{\lambda\in\mathbf{\Lambda}}$, we define the joint sparsity measure as the functional

$$\mathbf{\Psi}_{\mathcal{D}}^{(p)}(\mathbf{F}, v) := \sum_{\lambda\in\mathbf{\Lambda}} v_\lambda \, \|\mathbf{f}_\lambda\|_p + \sum_{\lambda\in\mathbf{\Lambda}} \omega_\lambda \|\mathbf{f}_\lambda\|_2^2 + \sum_{\lambda\in\mathbf{\Lambda}} \theta_\lambda (\rho_\lambda - v_\lambda)^2 \quad , \quad p \geq 1 \ . \tag{14}$$

By using the decomposition (9), the discrepancy (7) can be written as

$$\Delta(\mathbf{j}) = \|T\,\mathbf{j} - \mathbf{M}\|_{\mathbb{R}^N}^2 \quad , \tag{15}$$

where $\mathbf{j} := \left(j_\lambda^\ell\right)_{\lambda\in\Lambda, \ell=1,2,3}$ and the operator $T : \ell_2(\Lambda, \mathbb{R}^3) \to \mathbb{R}^N$ has entries given by

$$(T\,\mathbf{j})_l = \sum_{\ell=1}^{3} \sum_{\lambda\in\mathbf{\Lambda}} j_\lambda^\ell \, \frac{\mu_0}{4\pi} \int_{V_0} \left(\frac{\mathbf{e}_z(\mathbf{q}_l) \times (\mathbf{r}\,' - \mathbf{q}_l)}{|\mathbf{r}\,' - \mathbf{q}_l|^3} \right)_\ell \psi_\lambda(\mathbf{r}\,') \, d\mathbf{r}\,' \quad , \quad l = 1, \ldots, N \ . \tag{16}$$

Thus, the magnetic NDE inverse problem with the joint sparsity constraint consists in minimizing the functional

$$\begin{aligned} \mathcal{J}_{\theta,\omega,\rho}^{(p)}(\mathbf{j}, v) &= \|T\mathbf{j} - \mathbf{M}\|_{\mathbb{R}^N}^2 + \mathbf{\Psi}_{\mathcal{D}}(\mathbf{j}, v) = \\ &= \sum_{l=1}^{N} |(T\mathbf{j})_l - m_l|^2 + \sum_{\lambda\in\mathbf{\Lambda}} v_\lambda \, \|\mathbf{j}_\lambda\|_p + \sum_{\lambda\in\mathbf{\Lambda}} \omega_\lambda \|\mathbf{j}_\lambda\|_2^2 + \sum_{\lambda\in\mathbf{\Lambda}} \theta_\lambda (\rho_\lambda - v_\lambda)^2 \end{aligned} \tag{17}$$

jointly with respect to both $\mathbf{j}$ and v, restricted to $v_\lambda \geq 0$.

The minimization of the functional (17) promotes that all entries of the vector $\mathbf{j}_\lambda$ have the *same sparsity* pattern. Note that v serves as an indicator of large values of $\|\mathbf{j}_\lambda\|_p$ and $0 \leq v_\lambda \leq \rho_\lambda$, $\lambda \in \mathbf{\Lambda}$, at the minimum. The quadratic term $\sum_{\lambda\in\mathbf{\Lambda}} \omega_\lambda \|\mathbf{j}_\lambda\|_2^2$ makes the overall functional convex, depending on a suitable choice of the sequence $(\omega_\lambda)_{\lambda\in\mathbf{\Lambda}}$.

We remark that when $\theta_\lambda = 0$ and $\omega_\lambda = \alpha$ (a fixed constant) for all $\lambda \in \Lambda$, we obtain the usual Tikhonov regularization since $(v_\lambda)_{\lambda\in\Lambda} = 0$ in this case.

The convexity conditions for the functional $\mathcal{J}_{\theta,\rho,\omega}^{(p)}$ depend on the values of the parameters $(\theta_\lambda)_{\lambda\in\mathbf{\Lambda}}$, $(\rho_\lambda)_{\lambda\in\mathbf{\Lambda}}$, $(\omega_\lambda)_{\lambda\in\mathbf{\Lambda}}$. Details can be found in [11].

In particular, we recall that for $p = 1, 2, \infty$, $\mathcal{J}_{\theta,\rho,\omega}^{(p)}$ is convex if

$$(s_{\min} + \omega_\lambda)\,\theta_\lambda \geq \frac{\kappa_p}{4} = \begin{cases} \frac{3}{4} \text{ if } p = 1 \\ \\ \frac{1}{4} \text{ if } p = 2, \infty \end{cases} \tag{18}$$

where $s_{\min}$ is the minimum of the spectrum of T^*T. In case of strict inequality, $\mathcal{J}_{\theta,\rho,\omega}^{(p)}$ is strictly convex.

4 The Minimizing Algorithm

The minimizer $\mathbf{j}^*$ of the functional $\mathcal{J}^{(p)}_{\theta,\rho,\omega}$ can be approximated by the following thresholded Landweber algorithm, deduced from [12].

Algorithm 1.

$$\begin{cases} \text{Let } \gamma \text{ be a suitable relaxation parameter} \\ \text{Choose an arbitrary} \quad \mathbf{j}^{(0)} \in \ell_2(\mathbf{\Lambda}; \mathbb{R}^3) \\ \text{For} \quad 0 \le k \le K \quad \text{do} \quad \mathbf{j}^{(k+1)} = H^{(p)}_{\theta,\rho,\omega}\left(\mathbf{j}^{(k)} + \gamma\, T^*(M - T\,\mathbf{j}^{(k)})\right) \\ \text{Compute} \quad v^{(K+1)}_\lambda = \begin{cases} \rho_\lambda - \frac{1}{2\theta_\lambda}\|\mathbf{j}^{(K+1)}{}_\lambda\|_p & \text{if } \|\mathbf{j}^{(K+1)}_\lambda\|_p < 2\theta_\lambda\rho_\lambda \\ 0 & \text{otherwise} \end{cases} \qquad \lambda \in \mathbf{\Lambda} \end{cases}$$

(usually, the values $(v^{(K)}_\lambda)_{\lambda\in\mathbf{\Lambda}}$ are not needed).

The operator $H^{(p)}_{\theta,\rho,\omega}$ is a thresholding operator acting on a vector-valued function $\mathbf{f} \in \ell_2(\mathbf{\Lambda}; \mathbb{R}^3)$ as

$$\left(H^{(p)}_{\theta,\rho,\omega}(\mathbf{f})\right)_\lambda := (1+\omega_\lambda)^{-1} h^{(p)}_{\theta_\lambda(1+\omega_\lambda),\rho_\lambda}(\mathbf{f}_\lambda) \ , \quad \lambda \in \mathbf{\Lambda} \ . \tag{19}$$

The explicit expression of $h^{(p)}_{\theta,\rho}$ can be found in [12] where an efficient algorithm for its evaluation is also given.

Algorithm 2.

$$\begin{cases} \text{Choose a positive sequence} \quad v^{(0)} \\ \text{Let } \mathbf{f}^{(0)} = \mathbf{f}_\lambda \\ \text{For} \quad 0 \le k \le K \quad \text{do} \\ \qquad \mathbf{f}^{(k+1)} = S^{(p)}_{v^{(k)}}\left(\mathbf{f}^{(k)}\right) \\ \qquad v^{(k+1)}_\lambda = \begin{cases} \rho_\lambda - \frac{1}{2\theta_\lambda}\|\mathbf{f}^{(k+1)}\|_p & \text{if } \|\mathbf{f}^{(k+1)}\|_p < 2\theta_\lambda\rho_\lambda \\ 0 & \text{otherwise} \end{cases} \qquad \lambda \in \mathbf{\Lambda} \\ \text{Approximate} \quad h^{(p)}_{\theta,\rho}(\mathbf{f}_\lambda) \approx \mathbf{f}^{(K+1)} \end{cases}$$

The operator $S^{(p)}_v$ is itself a thresholding operator whose explicit expressions for $p = 1, 2$ are (cf. [11])

$$S^{(1)}_v(\mathbf{f}) = \left(S^{(1)}_v(\mathbf{f})\right)_{\ell=1,2,3} \tag{20}$$

where

$$\left(S_v^{(1)}(\mathbf{f})\right)_{\ell=1,2,3} = s_v^{(1)}(f_\ell) = \begin{cases} \operatorname{sign}(f_\ell)\left(|f_\ell| - \frac{v}{2}\right) & \text{if} \quad |f_\ell| > \frac{v}{2} \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

and

$$S_v^{(2)}(\mathbf{f}) = \begin{cases} \frac{\|\mathbf{f}\|_2 - v/2}{\|\mathbf{f}\|_2}\,\mathbf{f} & \text{if} \quad \|\mathbf{f}\|_2 > \frac{v}{2} \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

The convergence of Algorithm 1 to the minimizer of $\mathcal{J}^{(p)}_{\theta,\omega,\rho}$ can be proved as in [10].

Theorem 3. *Let $p = 1, 2, \infty$, and assume*

$$\inf_{\lambda\in\mathbf{\Lambda}} \theta_\lambda(s_{\min} + \omega_\lambda) > \frac{\kappa_p}{4} \tag{23}$$

with $s_{\min} := \min \mathrm{Sp}(T^*T)$. Then for any choice $\mathbf{j}^{(0)} \in \ell_2(\mathbf{\Lambda}; \mathbb{R}^3)$ the Algorithm 1 converges strongly to a fixed point $\mathbf{j}^* \in \ell_2(\mathbf{\Lambda}; \mathbb{R}^3)$ and the pair $(\mathbf{j}^*, v^*)$ with

$$v^*_\lambda = \begin{cases} \rho_\lambda - \frac{1}{2\theta_\lambda}\|\mathbf{j}^*\|_p & \text{if } \|\mathbf{j}^*\|_p < 2\theta_\lambda\rho_\lambda \\ 0 & \text{otherwise} \end{cases} \qquad \lambda \in \mathbf{\Lambda} \tag{24}$$

is the unique minimizer of $\mathcal{J}^{(p)}_{\theta,\rho,\omega}(\mathbf{j}, v)$. Moreover, we have the error estimate

$$\|\mathbf{j}^{(k)} - \mathbf{j}^*\|_2 \le \beta^k \|\mathbf{j}^{(0)} - \mathbf{j}^*\|_2 \ , \tag{25}$$

where $\beta := \sup_{\lambda\in\mathbf{\Lambda}} \frac{4\theta_\lambda(1-s_{\min})}{4\theta_\lambda(1+\omega_\lambda)-\kappa_p} < 1$.

The convergence of Algorithm 2 follows from [12, Prop. 3.4].

Now, we want to focus our attention on the term $T^*T\,\mathbf{j}^{(k)}$ which appears in the third line of Algorithm 1. The iterative procedure can be implemented only if this term can be efficiently approximated at each iteration step.

The explicit expression of $T^*T\,\mathbf{j}$ is given by (cf. [10])

$$(T^*T\mathbf{j})_{\lambda,\ell} = \sum_{\mu\in\mathbf{\Lambda}} \sum_{m=1}^{3} \left(\sum_{l=1}^{N} (A_{\ell,l}\psi_\lambda)(A_{m,l}\psi_\mu) \right) j^m_\mu \ , \qquad \lambda \in \mathbf{\Lambda} \ , \quad \ell = 1, 2, 3 \ , \tag{26}$$

where the operator $A_{\ell,l} : L_2(V_0; \mathbb{R}) \to \mathbb{R}$ is defined as

$$A_{\ell,l}\, f := \frac{\mu_0}{4\pi} \int_{V_0} \left(\frac{\mathbf{e}_z(\mathbf{q}_l) \times (\mathbf{r}\,' - \mathbf{q}_l)}{|\mathbf{r}\,' - \mathbf{q}_l|^3} \right)_\ell f(\mathbf{r}\,')\, d\mathbf{r}\,' \ . \tag{27}$$

Let $\mathcal{M}$ be the matrix whose entries are the coordinates of T^*T in the multiscale basis (ψ_λ), i.e.

$$\mathcal{M}_{(\lambda,\ell),(\mu,m)} := \sum_{l=1}^{N} (A_{\ell,l}\psi_\lambda)(A_{m,l}\psi_\mu) \ , \qquad \lambda \ , \mu \in \mathbf{\Lambda} \ , \quad \ell, m = 1, 2, 3 \ . \tag{28}$$

From (26) it follows

$$(T^*T\,\mathbf{j})_{\lambda,\ell} = \sum_{\mu\in\mathbf{\Lambda}}\sum_{m=1}^{3} j_\mu^m\,\mathcal{M}_{(\lambda,\ell),(\mu,m)}\ , \qquad \lambda\in\mathbf{\Lambda}\ , \quad \ell=1,2,3\ . \tag{29}$$

Since $\mathcal{M}$ is a bi-infinite matrix, in order to implement an efficient procedure to compute $T^*T\,\mathbf{j}^{(k)}$ we need the amplitude of the entries of $\mathcal{M}$ to decay fast when λ and μ increase.

Let us choose as multiscale basis a compactly supported wavelet basis with $\Omega_\lambda := \mathrm{supp}\,(\psi_\lambda) \sim 2^{-|\lambda|}$, where $|\lambda|$ denote the spatial resolution scale of ψ_λ. Moreover, we assume that the basis functions have d^* vanishing moments, a prescribed smoothness, and fast decay, i.e. $|\psi_\lambda| \le C2^{3/2|\lambda|}$.

It can be shown that $\mathcal{M}$ has compressibility properties w.r.t. such a basis (cf. [10], [18]), so that $T^*T\,\mathbf{j}^{(k)}$ can be evaluated with an efficient procedure. In particular, we have

$$|\mathcal{M}_{(\lambda,\ell),(\mu,m)}| \le C2^{-(|\lambda|+|\mu|)(3/2+d^*+1)}\times \\ \times \begin{cases} N\mathrm{dist}\,(\Omega_\lambda,\Omega_\mu)^{-(3+d^*)} & \text{if } \mathrm{dist}\,(\Omega_\lambda,\Omega_\mu)>0 \\ \displaystyle\sum_{l=1}^{N}(\mathrm{dist}\,(x_l,\Omega_\lambda)\mathrm{dist}\,(x_l,\Omega_\mu))^{-(3+d^*)} & \text{otherwise} \end{cases} \tag{30}$$

5 A Bidimensional Test Problem

We consider a bidimensional test problem and we assume the current is confined on an horizontal plane S_0, i.e.

$$\mathbf{J}(\mathbf{r}) = \big(J_1(x,y), J_2(x,y), 0\big)\ , \qquad (x,y)\in S_0\ . \tag{31}$$

By substituting (31) in (6) and integrating over z' we obtain, for $l=1,\dots,N$,

$$\mathcal{B}(\mathbf{J},\mathbf{q}_l) := \frac{\mu_0}{4\pi}d\int_{S_0}\frac{J_2(x',y')(x'-x_l)-J_1(x',y')(y'-y_l)}{\big((x'-x_l)^2+(y'-y_l)^2+z_l^2\big)^{3/2}}dx'\,dy'\ . \tag{32}$$

Since the current coordinates in the multiscale basis are bidimensional too, i.e.

$$\mathbf{j}_\lambda = (j_\lambda^1, j_\lambda^2, 0)\ , \qquad \lambda\in\mathbf{\Lambda}\ , \tag{33}$$

the entries of $T^*T\,\mathbf{j}$ become

$$(T^*T\mathbf{j})_{\lambda,\ell} = \sum_{\mu\in\mathbf{\Lambda}}\sum_{m=1}^{2}\left(\sum_{l=1}^{N}(A_{\ell,l}\psi_\lambda)(A_{m,l}\psi_\mu)\right)j_\mu^m\ , \qquad \lambda\in\mathbf{\Lambda}, \quad \ell=1,2\ , \tag{34}$$

with

$$A_{1,l}\,f := \frac{\mu_0}{4\pi}\int_{S_0}\frac{-(y'-y_l)}{|\mathbf{r}'-\mathbf{q}_l|^3}\,f(\mathbf{r}')\,d\mathbf{r}'\ , \quad A_{2,l}\,f := \frac{\mu_0}{4\pi}\int_{S_0}\frac{(x'-x_l)}{|\mathbf{r}'-\mathbf{q}_l|^3}\,f(\mathbf{r}')\,d\mathbf{r}'\ . \tag{35}$$

We assume there are $N = 400$ sensors located on a regular horizontal grid at a fixed height $\pi = 1\,mm$. In the numerical tests the measured magnetic field is generated by two sources, a square current loop of length $1\,cm$ and a current dipole on the center of the square, both located on S_0.

We choose as a basis the Daubechies orthonormal wavelets with $d^* = 4$ vanishing moments and discretize the plane S_0 with 64 pixels for each dimension. Finally, 3 multiresolution levels are used for the current decomposition.

We have realized a MATLAB code running on a x-64 based workstation with 8 processors. It takes about 25 min for the construction of the matrix $\mathcal{M}$ and about 300 sec for 120 iterations of the algorithm.

In Fig. 2 the distribution of the entries of the matrices $T_1^* T_1$ and $T_1^* T_2$ with amplitude greater than $10^{-4}(\|\mathcal{M}\|_\infty / \dim \mathcal{M}) \approx 0.1$ are displayed. Just about 20% of the entries are retained. The matrices $T_2^* T_1$ and $T_2^* T_2$ have a similar behaviour.

In Fig. 3 the measured magnetic field is displayed (the sensor locations are displayed as black points). In Fig. 4 the current distribution reconstructed after 120 iterations of the proposed algorithm is shown. The figures show that the proposed algorithm is able to saparate both the current sources. Actually, in this case the inverse problem is well-posed [2] and can be solved without a regularization procedure: indeed, the Tikhonov regularization with regularization parameter set equal to 0 gives a similar result.

In Fig. 6 the current distribution reconstructed after 30 iterations of Algorithm 1 is displayed when high white Gaussian noise with linear signal to noise ratio 1 is added to the magnetic field of Fig. 5. The thresholding parameter $(\rho_\lambda)_{\lambda \in \mathbf{\Lambda}}$ is chosen as

$$\rho_{|\lambda|} = \sigma \sqrt{\frac{2 \log(2^{|\lambda|} P)}{2^{|\lambda|} P}} \quad ,$$

where σ is the noise level, $|\lambda| = 0, \cdots, 3$, is the multiresolution scale, and $P = 64^2$ (cf. [7]). In the examples $\sigma \approx 133\,pT$, $\rho_0 \approx 8\,pT$. Finally, we set $(\omega_\lambda)_{\lambda \in \mathbf{\Lambda}} = 0.5$,

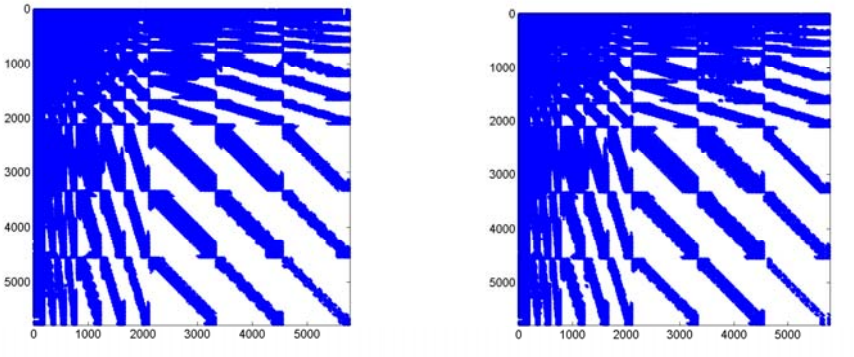


Fig. 2. The distribution of the entries of the matrices $T_1^* T_1$ (left) and $T_1^* T_2$ (right). Just the entries that exceed 0.1 are retained.

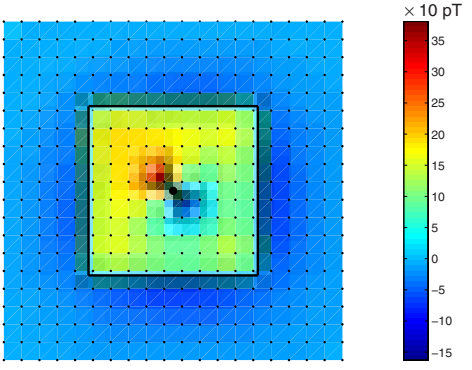


Fig. 3. The magnetic field (pT) produced by a square current loop (black square) with a current dipole on its center (black circle). The magnetometers are displayed as black points.

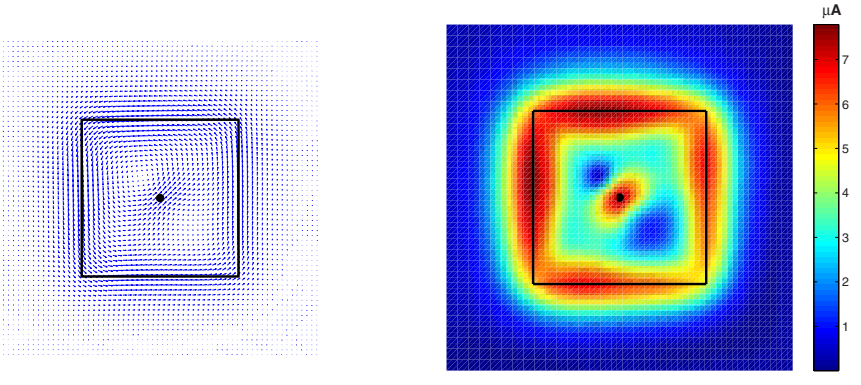


Fig. 4. The current distribution (left) reconstructed starting from the magnetic field displayed in Fig. 3. The current intensity (μA) is also displayed (right). The current loop and the current dipole are represented as a black square and a small circle, respectively.

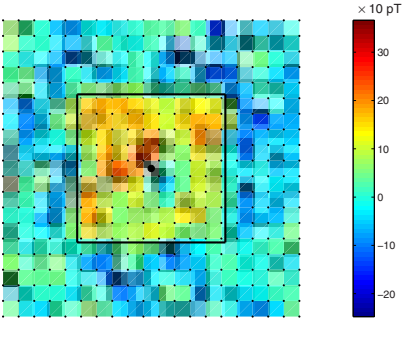


Fig. 5. The noisy magnetic field (pT) produced by a square current loop (black square) with a current dipole on its center (black circle). The magnetometers are displayed as black points.

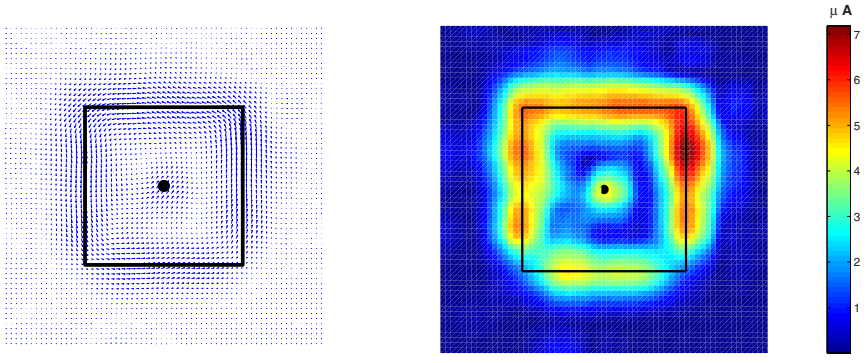


Fig. 6. The current distribution (left) reconstructed starting from the noisy magnetic field displayed in Fig. 5. The current intensity (μA) is also displayed (right). The current loop and the current dipole are represented as a black square and a small circle, respectively.

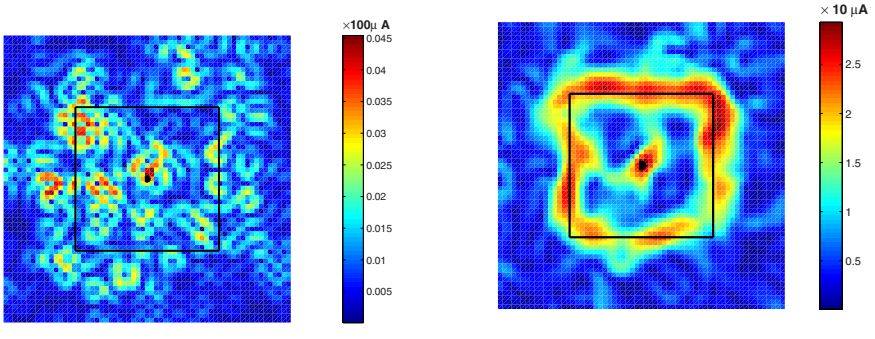


Fig. 7. The current distribution reconstructed by using the Tikhonov regularization starting from the noisy magnetic field displayed in Fig. 5. Two different values of the regularization parameter have been used: 0 (left) and 1001 (right). The current loop and the current dipole are represented as a black square and a small circle, respectively.

and $(\theta_\lambda)_{\lambda\in\mathbf{\Lambda}} = 1$. Nevertheless the reconstructed current images are less sharp than the images in the noiseless case, it is still possible to separate the two sources. On the contrary, Tikhonov regularization is not able to give a satisfactory current image. In fact, when the regularization parameter is chosen equal to 0 the current is not reconstructed at all (see Fig. 7, left), while a regularization parameter greater than 0 produces a distorted current image where the separation between the two sources is not sufficiently sharp (see Fig. 7, right, where the regularization parameter is chosen by means of the discrepancy principle and is equal to 1001).

6 Conclusion

The numerical tests in Section 5 show that the proposed algorithm outperforms Tikhonov regularization in the case when the magnetic field is affected by high noise. This behaviour would be more evident in the three-dimensional case, where the problem is highly ill-posed and ill-conditioned, and we expect our algorithm to give better results than the usual Tikhonov regularization.

In order to implement efficiently Algorithm 1 in the three-dimensional case we need to solve some practical problems. First of all, the dimension of the matrix approximating $\mathcal{M}$ dramatically increases and, moreover, accurate results require a high resolution, i.e. a high pixel number. This means that we have to improve compressibility properties of $\mathcal{M}$ by using a different, and better suited, multiscale basis.

Secondly, the convergence rate of the proposed algorithm can be very slow (see Fig. 8) and acceleration techniques are mandatory. At present acceleration techniques have been proposed in the scalar case [9], [16]. An extension to the three-dimensional case will be the subject of further investigations.

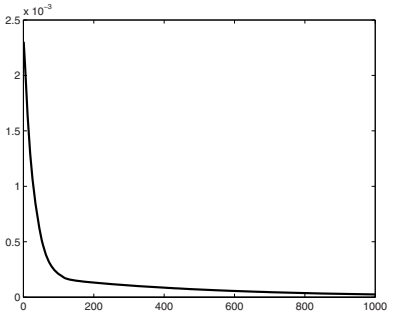


Fig. 8. The norm of the difference between two successive iterations for the first 1000 iterations of the proposed algorithm. The plot refers to the magnetic field measurements displayed in Fig. 4.

References

1. Bertero, M., Boccacci, P.: Introduction to inverse problems in imaging. Institute of Physics, Bristol, Philadelphia (2002)
2. Bradley, J.R., Wikswo, J.P., Sepulveda, N.G.: Using a magnetometer to image a two-dimensional current distribution. J. App. Phys. 65, 361–372 (1989)
3. Christensen, O.: An Introduction to Frames and Riesz Bases. Birkhäuser, Basel (2003)
4. Daubechies, I.: Ten Lectures on Wavelets. SIAM, Philadelphia (1992)
5. Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun. Pure Appl. Math. 57, 1413–1457 (2004)
6. Donoho, D.L.: Superresolution via sparsity constraints. SIAM J. Math. Anal. 23, 1309–1331 (1992)
7. Donoho, D.L.: De-noising by soft-thresholding. IEEE Trans. Inform. Theory 41, 613–627 (1995)

8. Engl, H.W., Hanke, M., Neubauer, A.: Regularization of inverse problems. Kluwer, Dordrecht (2000)
9. Daubechies, I., Fornasier, M., Loris, I.: Accelerated projected gradient methods for linear inverse problems with sparsity constraints. J. Fourier Anal. Appl. 14, 764–792 (2008)
10. Fornasier, M., Pitolli, F.: Adaptive iterative thresholding algorithms for magnetoencephalography (MEG). J. Comput. Appl. Math. 221, 386–395 (2008)
11. Fornasier, M., Rauhut, H.: Recovery algorithms for vector-valued data with joint sparsity constraints. SIAM J. Numer. Anal. 46, 577–613 (2008)
12. Fornasier, M., Rauhut, H.: Iterative thresholding algorithms. Appl. Comput. Harmon. Anal. 25, 187–208 (2008)
13. Jackson, J.D.: Classical Electrodynamics. John Wiley & Sons, Inc., Chichester (1975)
14. Kaipio, J., Somersalo, E.: Statistical and computational inverse problems. Applied Mathematical Sciences, vol. 160. Springer, New York (2005)
15. Kress, R., Kühn, L., Potthast, R.: Reconstruction of a current distribution from its magnetic field. Inverse Problems 18, 1127–1146 (2002)
16. Ramlau, R., Teschke, G., Zhariy, M.: A compressive Landweber iteration for solving ill-posed inverse problems. Inverse Problems 24, 1–26 (2008)
17. Sarvas, J.: Basic mathematical and electromagnetic concepts of the biomagnetic inverse problem. Phys. Med. Biol. 32, 11–22 (1987)
18. Schneider, R.: Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur effizienten Lösung grosser vollbesetzter Gleichungssysteme. Advances in Numerical Mathematics, Teubner, Stuttgart (1998)
19. Wikswo, J.P.: Applications of SQUID magnetometers to biomagnetism and nondestructive evaluation. In: Weinstock, H. (ed.) Applications of Superconductivity, pp. 139–228. Kluwer Academic Publishers, Netherlands (2000)

Constructing Good Coefficient Functionals for Bivariate C^1 Quadratic Spline Quasi-Interpolants

Sara Remogna

Università degli Studi di Torino, Dipartimento di Matematica
Via Carlo Alberto 10, 10123 Torino, Italy
sara.remogna@unito.it

Abstract. We consider discrete quasi-interpolants based on C^1 quadratic box-splines on uniform criss-cross triangulations of a rectangular domain. The main problem consists in finding good (if not best) coefficient functionals, associated with boundary box-splines, giving both an optimal approximation order and a small infinity norm of the operator. Moreover, we want that these functionals only involve data points inside the domain. They are obtained either by minimizing their infinity norm w.r.t. a finite number of free parameters, or by inducing superconvergence of the operator at some specific points lying near or on the boundary.

1 Introduction

Let $\Omega = [0, hm] \times [0, hn]$ be a rectangular domain divided into mn equal squares, where $m, n \geq 4$, each of them being subdivided into 4 triangles by its diagonals. We denote by $S_2^1(\Omega, \mathscr{T})$ the space of C^1 quadratic splines on the triangulation $\mathscr{T}$ of Ω obtained in this way. This space is generated by the $(m+2)(n+2)$ spline functions $\{B_\alpha, \alpha \in A\}$, where $A = \{(i, j), 0 \leq i \leq m+1, 0 \leq j \leq n+1\}$, obtained by dilation/translation of the Zwart-Powell quadratic box-spline (ZP-element) ([5] Chap.1, [6] Chap.3).

The ZP-element is the bivariate C^1 quadratic box-spline supported on the octagon with center at the origin and vertices at $(\frac{3}{2}, \frac{1}{2})$, $(\frac{1}{2}, \frac{3}{2})$, $(-\frac{1}{2}, \frac{3}{2})$, $(-\frac{3}{2}, \frac{1}{2})$, $(-\frac{3}{2}, -\frac{1}{2})$, $(-\frac{1}{2}, -\frac{3}{2})$, $(\frac{1}{2}, -\frac{3}{2})$, $(\frac{3}{2}, -\frac{1}{2})$. It is strictly positive inside its support, that is partitioned into 28 triangular cells. On every cell the ZP-element is a polynomial of total degree 2 and in [7] the polynomials are given. The ZP-element can be also expressed in BB-form, i.e. specifying the Bernstein-Bézier (abbr. BB)-coefficients on every triangular cell, ([15] Chap.6).

In the space $S_2^1(\Omega, \mathscr{T})$ we consider discrete quasi-interpolants (abbr. dQI) of type

$$Qf = \sum_{\alpha \in A} \lambda_\alpha(f) B_\alpha, \tag{1}$$

where $\{\lambda_\alpha,\ \alpha \in A\}$ is a family of linear functionals which are local, in the sense that they are linear combinations of values of f at some points lying inside Ω and in a neighbourhood of the support Σ_α of B_α and such that Q is exact on the space $\mathbb{P}_2$ of quadratic polynomials.

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 329–346, 2010.

The points M_α, $\alpha \in A$, used in evaluating f, are the centers of the squares (mn points), the midpoints of boundary segments ($2(m+n)$ points) and the four vertices of Ω, see Fig. 1. These points are defined by $M_\alpha = M_{i,j} = (s_i, t_j)$ where

$$\begin{aligned} &s_0 = 0,\ s_i = (i - \tfrac{1}{2})h,\ 1 \le i \le m,\ s_{m+1} = hm, \\ &t_0 = 0,\ t_j = (j - \tfrac{1}{2})h,\ 1 \le j \le n,\ t_{n+1} = hn. \end{aligned} \tag{2}$$

The values of the function f at those points are denoted by $f_\alpha = f(M_\alpha)$.

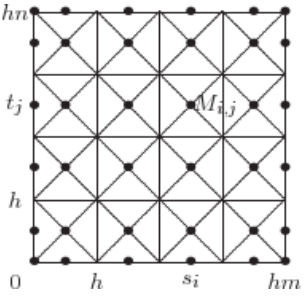


Fig. 1. Uniform criss-cross triangulation and data points

Another spanning set for the space $S_2^1(\Omega, \mathscr{T})$ is formed by the box-splines $\{B'_\alpha, \alpha \in A\}$ constructed in [9] and [19], but, in some applications, it may be more convenient to make use of classical box-splines instead of the other ones, because the latter have different supports and different expressions in the domain, while the ZP-element is always the same.

The quasi-interpolants constructed in this paper are suitable for some applications. For example they can be used for the approximation of the gradient of a function and the computation of critical points and curvatures of a surface [12]. Furthermore they can be used in the second stage of the two-stage method of Schumaker [21].

Here is an outline of the paper. In Section 2 we compare different choices for the functionals of such dQIs: near-best functionals and functionals inducing superconvergence at specific points. In Section 3 we estimate the operator norm by estimating the Lebesgue function near the origin of Ω for two particular dQIs. Finally, in Section 4 we give some numerical and graphical results.

2 On the Construction of Coefficient Functionals

In this section we are interested in the construction of linear functionals

$$\lambda_\alpha(f) = \sum_{\beta \in F_\alpha} \sigma_\alpha(\beta) f(M_\beta), \tag{3}$$

where the finite set of points $\{M_\beta,\ \beta \in F_\alpha\}$, $F_\alpha \subset A$, lies in some neighbourhood of $\Sigma_\alpha \cap \Omega$ and such that $Qf \equiv f$ for all f in $\mathbb{P}_2$.

The construction of such coefficient functionals is related to the following differential QI (abbr. DQI), exact on $\mathbb{P}_2$ and defined on the infinite plane, $A = \mathbb{Z}^2$ ([15] Chap.6):

$$\widehat{Q}f = \sum_{\alpha \in A} \left(f_\alpha - \frac{h^2}{8} \Delta f_\alpha \right) B_\alpha. \tag{4}$$

By using the five point discretisation of the Laplacian Δ, from (4), the following discrete functionals $\lambda_{i,j}$ are defined in [16] (see also [17])

$$\lambda_{i,j}(f) = \frac{3}{2} f_{i,j} - \frac{1}{8}(f_{i-1,j} + f_{i+1,j} + f_{i,j-1} + f_{i,j+1}). \tag{5}$$

In the interior of the domain our quasi-interpolants make use of the same inner functionals $\lambda_{i,j}$, with $i = 2, \ldots, m-1$, $j = 2, \ldots, n-1$, defined by (5).

For the sake of simplicity, we assume $h = 1$ and we propose two different ways of constructing functionals associated with the box splines whose supports are not entirely inside Ω: near-best functionals (denoted by λ'_α and λ''_α) and functionals (denoted by λ_α) inducing superconvergence at some specific points.

2.1 Near-Best Boundary Functionals

In this section we construct convenient boundary coefficient functionals, called *near-best functionals*, giving both the best approximation order and a small infinity norm of the operator. In the definition of functionals if we consider more data points than the number of conditions we are imposing, we obtain a system of equations with free parameters. We choose them by minimizing their infinity norm.

The method used in this subsection is closely related to the techniques given in [1,2, 4,14] to define near-best discrete quasi-interpolants on type-1 and type-2 triangulations (see also [3,18]).

From (3) it is clear that, for $\|f\|_\infty \leq 1$ and $\alpha \in A$, $|\lambda_\alpha(f)| \leq \|\sigma_\alpha\|_1$ where σ_α is the vector with components $\sigma_\alpha(\beta)$, from which we deduce immediately

$$|Qf| \leq \sum_{\alpha \in A} |\lambda_\alpha(f)| B_\alpha \leq \max_{\alpha \in A} |\lambda_\alpha(f)| \leq \max_{\alpha \in A} \|\sigma_\alpha\|_1,$$

therefore we can conclude

$$\|Q\|_\infty \leq \max_{\alpha \in A} \|\sigma_\alpha\|_1.$$

Now assuming that $\mathrm{card}(F_\alpha) > 6$, we can try to find $\sigma^*_\alpha \in \mathbb{R}^{\mathrm{card}(F_\alpha)}$ solution of the minimization problem (see e.g. [4], [14] Chap.3)

$$\|\sigma^*_\alpha\|_1 = \min \left\{ \|\sigma_\alpha\|_1 ; \sigma_\alpha \in \mathbb{R}^{\mathrm{card}(F_\alpha)}, V_\alpha \sigma_\alpha = b_\alpha \right\},$$

where $V_\alpha \sigma_\alpha = b_\alpha$ is a linear system expressing that Q is exact on $\mathbb{P}_2$. In our case we require that the coefficient functional coincides with the differential coefficient $(f - \frac{1}{8}\Delta f)$ at the center c_α of the octagonal support of the box-spline B_α, for $f \in \mathbb{P}_2$.

This problem is a l_1-minimization problem and there are many well-known techniques for approximating the solutions, not unique in general (cf. [22] Chap.6). Since the minimization problem is equivalent to a linear programming one, here we use the simplex method.

Hereinafter we analyse some coefficient functionals λ'_α near the boundary of the domain Ω.

Case $\alpha = (0,0)$

We consider the 10-point linear functional

$$\lambda'_{0,0}(f) = a_1 f_{0,0} + a_2(f_{1,0} + f_{0,1}) + a_3(f_{2,0} + f_{0,2}) + a_4(f_{3,0} + f_{0,3}) + a_5 f_{1,1} + a_6(f_{2,1} + f_{1,2}),$$

and we impose $\lambda'_{0,0}(f) \equiv (f - \frac{1}{8}\Delta f)(c_{00})$, $c_{00} = (-\frac{1}{2}, -\frac{1}{2})$, for $f \equiv 1, x, x^2, xy$. Due to symmetry of the ZP-element, there are only 6 unknowns and the monomials y and y^2 can be excluded.

This leads to the system:

$$a_1 + 2a_2 + 2a_3 + 2a_4 + a_5 + 2a_6 = 1,\ a_2 + 3a_3 + 5a_4 + a_5 + 4a_6 = -1,$$
$$a_2 + 9a_3 + 25a_4 + a_5 + 10a_6 = 0,\ a_5 + 6a_6 = 1,$$

whose solution depends on the two parameters a_4 and a_6

$$a_1 = \frac{14}{3} - \frac{16}{3}a_4 - 4a_6,\ a_2 = -\frac{5}{2} + 5a_4 + 5a_6,\ a_3 = \frac{1}{6} - \frac{10}{3}a_4 - a_6,\ a_5 = 1 - 6a_6.$$

If we minimize the norm $\|\lambda'_{0,0}\|_\infty$ we obtain

$$a_1 = \frac{20}{9},\ a_2 = 0,\ a_3 = -\frac{10}{9},\ a_4 = \frac{1}{3},\ a_5 = 0,\ a_6 = \frac{1}{6},$$

with a norm equal to $\|\lambda'_{0,0}\|_\infty = 5.44$. Graphically we obtain a functional with data points shown in Fig.2 (left).

If we want a functional with a smaller norm, we can add e.g. the point M_{22}

$$\lambda''_{0,0}(f) = a_1 f_{0,0} + a_2(f_{1,0} + f_{0,1}) + a_3(f_{2,0} + f_{0,2}) + a_4(f_{3,0} + f_{0,3}) + a_5 f_{1,1} + a_6(f_{2,1} + f_{1,2}) + a_7 f_{2,2}.$$

Solving the corresponding system and minimizing the norm $\|\lambda''_{0,0}\|_\infty$, we obtain

$$a_1 = \frac{22}{9},\ a_2 = 0,\ a_3 = -\frac{1}{9},\ a_4 = 0,\ a_5 = -\frac{3}{2},\ a_6 = 0,\ a_7 = \frac{5}{18},$$

with a norm equal to $\|\lambda''_{0,0}\|_\infty = 4.44$. Graphically we obtain the functional with data points shown in Fig.2 (right).

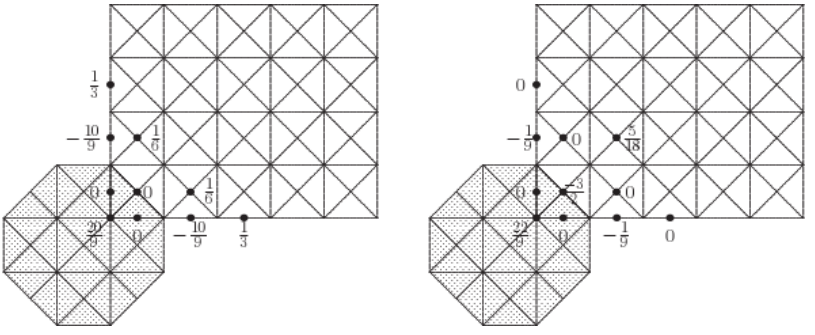


Fig. 2. Near-best functionals associated with $B_{0,0}$

Case $\alpha = (1,0)$

We consider

$$\lambda'_{1,0}(f) = a_1 f_{0,0} + a_2 f_{1,0} + a_3 f_{0,1} + a_4 f_{2,0} + a_5 f_{0,2} + a_6 f_{3,0} + a_7 f_{0,3} + a_8 f_{1,1} + a_9 f_{2,1} + a_{10} f_{1,2}.$$

Solving the corresponding system and minimizing the norm we obtain

$$a_1 = \tfrac{38}{45},\ a_2 = a_3 = 0,\ a_4 = \tfrac{19}{18},\ a_5 = -\tfrac{1}{2},\ a_6 = -\tfrac{11}{30},$$
$$a_7 = \tfrac{3}{10},\ a_8 = a_9 = 0,\ a_{10} = -\tfrac{1}{3},$$

with a norm equal to $\|\lambda'_{1,0}\|_\infty = 3.4$, see Fig. 3.

Case $\alpha = (2,0)$

We consider

$$\lambda'_{2,0}(f) = a_1 f_{2,0} + a_2(f_{1,0} + f_{3,0}) + a_3 f_{2,1} + a_4(f_{1,1} + f_{3,1}) + a_5 f_{2,2} + a_6(f_{1,2} + f_{3,2}).$$

Solving the corresponding system and minimizing the norm we obtain

$$a_1 = \frac{7}{3},\ a_2 = 0,\ a_3 = -\frac{13}{12},\ a_4 = -\frac{5}{24},\ a_5 = 0,\ a_6 = \frac{1}{12},$$

with a norm equal to $\|\lambda'_{2,0}\|_\infty = 4$, see Fig. 4 (left).

If we add the point M_{23},

$$\lambda''_{2,0}(f) = a_1 f_{2,0} + a_2(f_{1,0} + f_{3,0}) + a_3 f_{2,1} + a_4(f_{1,1} + f_{3,1}) + a_5 f_{2,2} + a_6(f_{1,2} + f_{3,2}) + a_7 f_{2,3},$$

we get

$$a_1 = \frac{23}{15},\ a_2 = a_3 = a_4 = 0,\ a_5 = -\frac{7}{12},\ a_6 = -\frac{1}{8},\ a_7 = \frac{3}{10},$$

with a norm equal to $\|\lambda''_{2,0}\|_\infty = 2.67$, see Fig. 4 (right).

Case $\alpha = (1,1)$

We consider

$$\lambda'_{1,1}(f) = a_1 f_{1,1} + a_2(f_{2,1} + f_{1,2}) + a_3(f_{3,1} + f_{1,3}) + a_4 f_{2,2}.$$

Solving the linear system we obtain

$$a_1 = \frac{3}{4},\ a_2 = \frac{1}{4},\ a_3 = -\frac{1}{8},\ a_4 = 0,$$

with a norm equal to $\|\lambda'_{1,1}\|_\infty = 1.5$, see Fig. 5 (left).

Case $\alpha = (2,1)$

We consider

$$\begin{aligned}\lambda'_{2,1}(f) = {} & a_1 f_{2,1} + a_2(f_{1,1} + f_{3,1}) + a_3 f_{2,2} + a_4(f_{1,2} + f_{3,2}) \\ & + a_5 f_{2,3} + a_6(f_{1,3} + f_{3,3}).\end{aligned}$$

Solving the corresponding system and minimizing the norm we obtain

$$a_1 = \frac{7}{8},\ a_2 = 0,\ a_3 = \frac{3}{8},\ a_4 = -\frac{1}{16},\ a_5 = 0,\ a_6 = -\frac{1}{16},$$

with a norm equal to $\|\lambda'_{2,1}\|_\infty = 1.5$, see Fig. 5 (right).

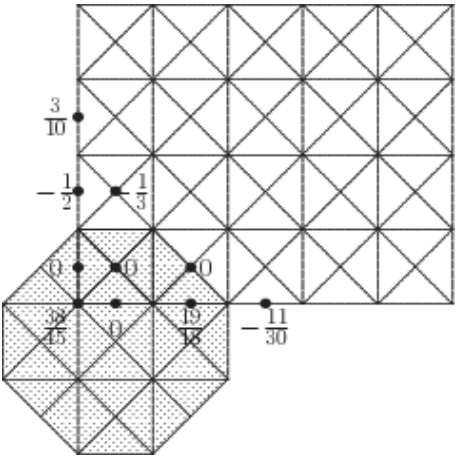


Fig. 3. Near-best functional associated with $B_{1,0}$

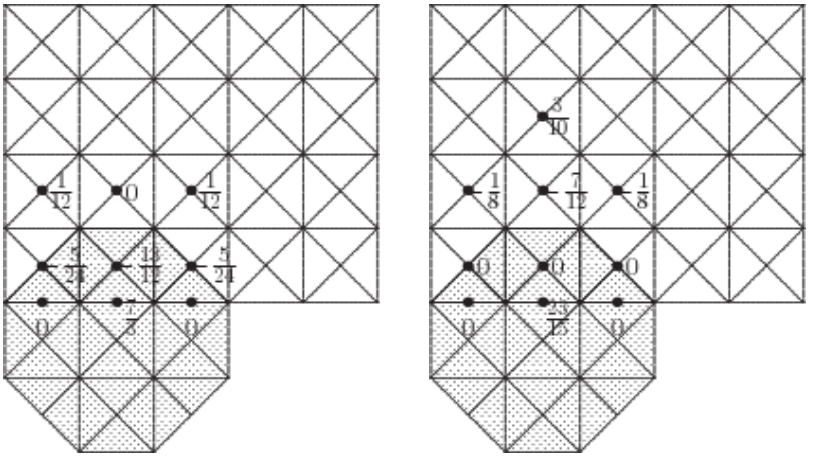


Fig. 4. Near-best functionals associated with $B_{2,0}$

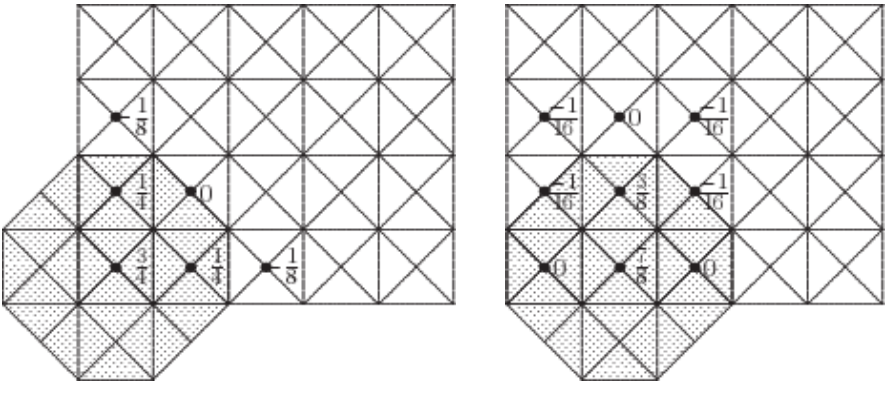


Fig. 5. Near-best functionals associated with $B_{1,1}$ (left) and $B_{2,1}$ (right)

2.2 Boundary Functionals Inducing Superconvergence

In this section we construct boundary coefficient functionals inducing superconvergence of the operator Q at some specific points. Using the notations given in (2) these specific points are (see Fig.6):

- the vertices of squares $A_{k,l} = (kh, lh)$, $k = 0, \ldots, m$, $l = 0, \ldots, n$,
- the centers of squares $M_{k,l} = (s_k, t_l)$, $k = 1, \ldots, m$, $l = 1, \ldots, n$,
- the midpoints $C_{k,l} = (s_k, lh)$ of horizontal edges $A_{k-1,l}A_{k,l}$, $k = 1, \ldots, m$, $l = 0, \ldots, n$,
- the midpoints $D_{k,l} = (kh, t_l)$ of vertical edges $A_{k,l-1}A_{k,l}$, $k = 0, \ldots, m$, $l = 1, \ldots, n$.

We remark that if $A = \mathbb{Z}^2$ the discrete operator defined by using the coefficient functionals given in (5) is superconvergent at these points [11,20].

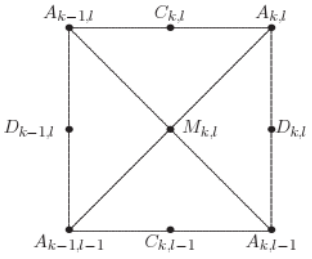


Fig. 6. A square of the uniform criss-cross triangulation

Recall that we construct the boundary coefficient functionals $\lambda_\alpha(f)$ so that they coincide with the differential coefficients $(f-\frac{1}{8}\Delta f)$ at the center c_α of the octagonal support of the box-splines B_α for $f\in\mathbb{P}_2$.

Since the differential quasi-interpolant (4) is exact on $\mathbb{P}_2$, the discrete operator that we are constructing is also exact on $\mathbb{P}_2$, therefore the approximation order $f-Qf$ is $\mathcal{O}(h^3)$ for smooth functions.

If we want superconvergence at some specific points, i.e. $f(M)-Qf(M)=\mathcal{O}(h^4)$, we have to require that, for $f\in\mathbb{P}_3$, the quasi-interpolant Q interpolates the function f at those points. So we impose that $Qf(M)=f(M)$ for $f\in\mathbb{P}_3\backslash\mathbb{P}_2$, M being a specific point of the domain.

This leads to a system of equations. We consider systems with free parameters and we choose them by minimizing the infinity norm $\|\lambda_\alpha\|_\infty$ and solving the corresponding l_1-minimization problem. Hereinafter we analyse some coefficient functionals λ_α near the boundary of Ω.

Case $\alpha=(2,1)$

We consider the 12-point linear functionals (defined using 8 unknowns)

$$\lambda_{2,1}(f)=a_1f_{2,1}+a_2(f_{1,1}+f_{3,1})+a_3f_{2,2}+a_4(f_{1,2}+f_{3,2})+a_5f_{2,3}+\\ +a_6(f_{1,3}+f_{3,3})+a_7f_{2,0}+a_8(f_{1,0}+f_{3,0}).$$

We require that:

(i) $\lambda_{2,1}(f)$ coincides with the differential coefficients $(f-\frac{1}{8}\Delta f)(c_{2,1})$, $c_{2,1}=(\frac{3}{2},\frac{1}{2})$ for $f\in\mathbb{P}_2$, i.e. for $f\equiv 1,x,y,x^2,xy,y^2$;

(ii) $Qf(M)=f(M)$ for $f\in\mathbb{P}_3\backslash\mathbb{P}_2$, i.e. for $f\equiv x^3,x^2y,xy^2,y^3$, and $M=(2,1)$, $(\frac{3}{2},\frac{3}{2})$, $(2,\frac{3}{2})$.

This leads to a system whose solution depends on the two parameters a_4 and a_6

$$a_1=\tfrac{15}{8}+6a_4+10a_6,\ \ a_2=-\tfrac{1}{8}-3a_4-5a_6,\ \ a_3=-\tfrac{1}{4}-2a_4,\\ a_5=\tfrac{1}{40}-2a_6,\ \ a_7=-\tfrac{2}{5}-8a_6-4a_4,\ \ a_8=2a_4+4a_6.$$

If we minimize the norm $\|\lambda_{2,1}\|_\infty$ we obtain

$$a_1 = \tfrac{13}{8},\ a_2 = 0,\ a_3 = -\tfrac{1}{6},\ a_4 = -\tfrac{1}{24},\ a_5 = \tfrac{1}{40},$$
$$a_6 = 0,\ a_7 = -\tfrac{7}{30},\ a_8 = -\tfrac{1}{12},$$

with a norm equal to $\|\lambda_{2,1}\|_\infty = 2.3$. Graphically we obtain a functional with data points shown in Fig. 7.

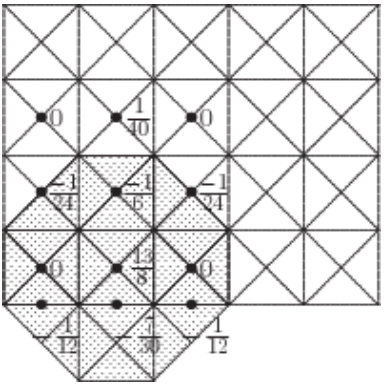


Fig. 7. Functional inducing superconvergence associated with $B_{2,1}$

Case $\alpha = (1,1)$

We consider

$$\lambda_{1,1}(f) = a_1 f_{0,0} + a_2(f_{1,0} + f_{0,1}) + a_3 f_{1,1} + a_4(f_{2,0} + f_{0,2}) + a_5(f_{2,1} + f_{1,2})$$
$$+ a_6(f_{3,0} + f_{0,3}) + a_7 f_{2,2} + a_8(f_{3,1} + f_{1,3}).$$

We require that:

(i) $\lambda_{1,1}(f)$ coincides with the differential coefficients $(f - \frac{1}{8}\Delta f)(\frac{1}{2}, \frac{1}{2})$ for $f \in \mathbb{P}_2$;
(ii) $Qf(M) = f(M)$ for $f \in \mathbb{P}_3 \backslash \mathbb{P}_2$ and $M = (1,1),\ (\frac{3}{2}, 1)$.

Solving the corresponding system and minimizing the norm we obtain

$$a_1 = -\tfrac{4}{15},\ a_2 = 0,\ a_3 = \tfrac{33}{20},\ a_4 = -\tfrac{2}{15},$$
$$a_5 = -\tfrac{1}{20},\ a_6 = 0,\ a_7 = -\tfrac{1}{15},\ a_8 = \tfrac{1}{40},$$

with a norm equal to $\|\lambda_{1,1}\|_\infty = 2.4$, see Fig. 8 (left).

Case $\alpha = (2,0)$

We consider

$$\lambda_{2,0}(f) = a_1 f_{2,1} + a_2(f_{1,1} + f_{3,1}) + a_3 f_{2,2} + a_4(f_{1,2} + f_{3,2}) + a_5 f_{2,3}$$
$$+ a_6(f_{1,3} + f_{3,3}) + a_7 f_{2,0} + a_8(f_{1,0} + f_{3,0}).$$

We require that:

(i) $\lambda_{2,0}(f)$ coincides with the differential coefficients $(f-\frac{1}{8}\Delta f)(\frac{3}{2},-\frac{1}{2})$ for $f\in\mathbb{P}_2$;
(ii) $Qf(M)=f(M)$ for $f\in\mathbb{P}_3\backslash\mathbb{P}_2$ and $M=(2,0),\ (\frac{3}{2},\frac{1}{2}),\ (2,\frac{1}{2})$.

Solving the corresponding system and minimizing the norm we obtain

$$a_1=-\frac{9}{8},\ a_2=-\frac{1}{4},\ a_3=0,\ a_4=\frac{1}{8},\ a_5=-\frac{1}{40},\ a_6=0,\ a_7=\frac{12}{5},\ a_8=0,$$

with a norm equal to $\|\lambda_{2,0}\|_\infty=4.3$, see Fig. 8 (right).

Case $\alpha=(1,0)$

We consider

$$\begin{aligned}\lambda_{1,0}(f)=\ & a_1f_{0,0}+a_2f_{1,0}+a_3f_{2,0}+a_4f_{3,0}+a_5f_{0,1}+a_6f_{1,1}+a_7f_{2,1}\\ & +a_8f_{0,2}+a_9f_{1,2}+a_{10}f_{0,3}+a_{11}f_{4,0}+a_{12}f_{3,1}+a_{13}f_{2,2}+a_{14}f_{1,3}.\end{aligned}$$

We require that:

(i) $\lambda_{1,0}(f)$ coincides with the differential coefficients $(f-\frac{1}{8}\Delta f)(\frac{1}{2},-\frac{1}{2})$ for $f\in\mathbb{P}_2$;
(ii) $Qf(M)=f(M)$ for $f\in\mathbb{P}_3\backslash\mathbb{P}_2$ and $M=(1,0),\ (\frac{1}{2},\frac{1}{2}),\ (1,\frac{1}{2})$.

Solving the corresponding system and minimizing the norm we obtain

$$\begin{gathered}a_1=0,\ a_2=\tfrac{131}{60},\ a_3=\tfrac{9}{40},\ a_4=0,\ a_5=-\tfrac{173}{300},\ a_6=-\tfrac{13}{40},\ a_7=-\tfrac{47}{60},\\ a_8=a_9=0,\ a_{10}=\tfrac{3}{20},\ a_{11}=-\tfrac{1}{120},\ a_{12}=\tfrac{3}{50},\ a_{13}=\tfrac{1}{4},\ a_{14}=-\tfrac{7}{40},\end{gathered}$$

with a norm equal to $\|\lambda_{1,0}\|_\infty=4.74$, see Fig. 9 (left).

Case $\alpha=(0,0)$

We consider

$$\begin{aligned}\lambda_{0,0}(f)=\ & a_1f_{0,0}+a_2(f_{1,0}+f_{0,1})+a_3f_{1,1}+a_4(f_{2,0}+f_{0,2})+a_5(f_{2,1}+f_{1,2})\\ & +a_6(f_{3,0}+f_{0,3})+a_7f_{2,2}+a_8(f_{3,1}+f_{1,3})+a_9(f_{4,0}+f_{0,4})\\ & +a_{10}(f_{4,1}+f_{1,4})+a_{11}(f_{2,3}+f_{3,2}).\end{aligned}$$

We require that:

(i) $\lambda_{0,0}(f)$ coincides with the differential coefficients $(f-\frac{1}{8}\Delta f)(-\frac{1}{2},-\frac{1}{2})$ for $f\in\mathbb{P}_2$;
(ii) $Qf(M)=f(M)$ for $f\in\mathbb{P}_3\backslash\mathbb{P}_2$ and $M=(0,0),\ (\frac{1}{2},0)$.

Solving the corresponding system and minimizing the norm we obtain

$$\begin{gathered}a_1=\tfrac{1403}{504},\ a_2=0,\ a_3=-\tfrac{63}{32},\ a_4=-\tfrac{397}{1440},\ a_5=a_6=0,\ a_7=\tfrac{317}{288},\\ a_8=0,\ a_9=\tfrac{11}{224},\ a_{10}=0,\ a_{11}=-\tfrac{37}{160},\end{gathered}$$

with a norm equal to $\|\lambda_{0,0}\|_\infty=7.0$, see Fig. 9 (right).

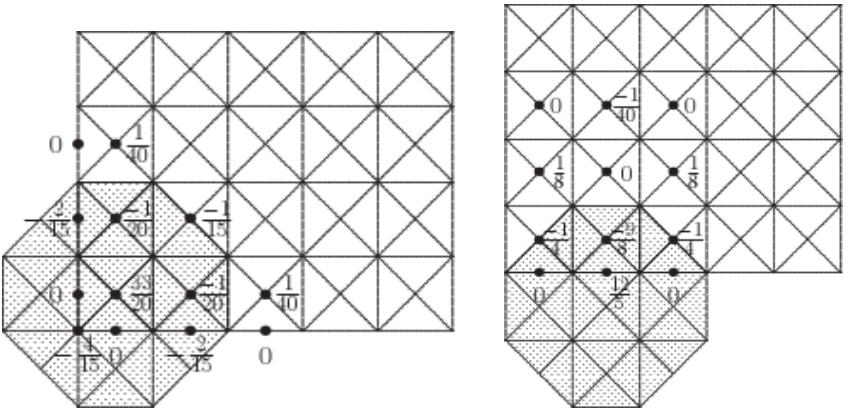


Fig. 8. Functionals inducing superconvergence associated with $B_{1,1}$ (left) and $B_{2,0}$ (right)

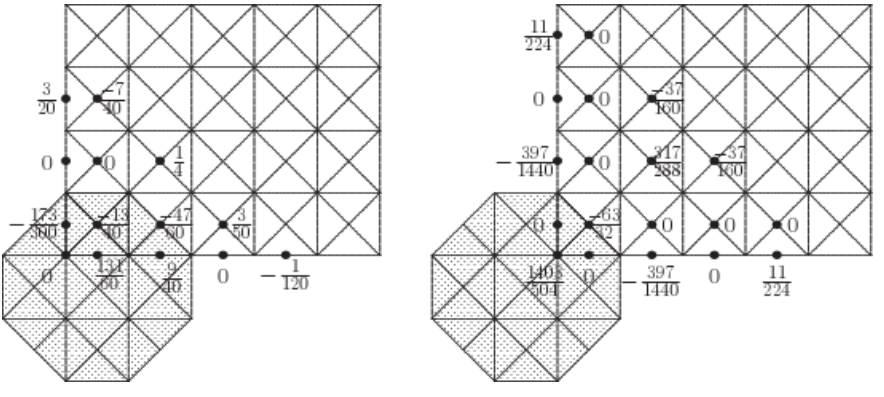


Fig. 9. Functionals inducing superconvergence associated with $B_{1,0}$ (left) and $B_{0,0}$ (right)

3 Two Examples of Discrete Quasi-Interpolants

Now, choosing conveniently the boundary coefficient functionals, we define two different discrete QIs of type (1) and we study their infinity norm on some square cells of the domain Ω, as shown in Fig. 10. We recall that in the interior of the domain the coefficient functionals $\lambda_{i,j}$, with $i = 2, \ldots, m-1$, $j = 2, \ldots, n-1$ are defined in (5).

3.1 The Discrete Quasi-Interpolant $Q_1 f$

A first discrete QI is defined by

$$Q_1 f = \sum_{\alpha \in A} \lambda_\alpha(f) B_\alpha,$$

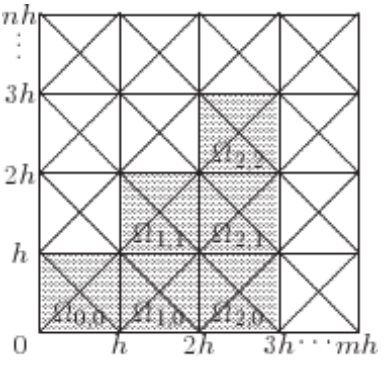


Fig. 10. Some square cells near the origin

where the boundary functionals are near-best in the sense of Sect. 2.1. Thanks to symmetry properties, only functionals in the neighbourhood of the origin are required:

$$\begin{aligned}
\lambda_{0,0}(f) &= \tfrac{22}{9} f_{0,0} - \tfrac{1}{9}(f_{2,0} + f_{0,2}) - \tfrac{3}{2} f_{1,1} + \tfrac{5}{18} f_{2,2}, \\
\lambda_{1,0}(f) &= \tfrac{38}{45} f_{0,0} + \tfrac{19}{18} f_{2,0} - \tfrac{1}{2} f_{0,2} - \tfrac{11}{30} f_{3,0} + \tfrac{3}{10} f_{0,3} - \tfrac{1}{3} f_{1,2}, \\
\lambda_{1,1}(f) &= \tfrac{3}{4} f_{1,1} + \tfrac{1}{4}(f_{2,1} + f_{1,2}) - \tfrac{1}{8}(f_{3,1} + f_{1,3}).
\end{aligned}$$

Along the lower edge, for $i = 2, \ldots, m-1$, we have

$$\begin{aligned}
\lambda_{i,0}(f) &= \tfrac{23}{15} f_{i,0} - \tfrac{7}{12} f_{i,2} - \tfrac{1}{8}(f_{i-1,2} + f_{i+1,2}) + \tfrac{3}{10} f_{i,3}, \\
\lambda_{i,1}(f) &= \tfrac{7}{8} f_{i,1} + \tfrac{3}{8} f_{i,2} - \tfrac{1}{16}(f_{i-1,2} + f_{i+1,2}) - \tfrac{1}{16}(f_{i-1,3} + f_{i+1,3}),
\end{aligned}$$

and analogous formulae along the three other edges of Ω.

3.2 The Discrete Quasi-Interpolant $Q_2 f$

A second discrete QI is defined by

$$Q_2 f = \sum_{\alpha \in A} \mu_\alpha(f) B_\alpha,$$

where we choose boundary functionals inducing superconvergence at the specific points defined in Sect. 2.2. Thanks to symmetry properties, only functionals in the neighbourhood of the origin are required:

$$\begin{aligned}
\mu_{0,0}(f) &= \tfrac{1403}{504} f_{0,0} - \tfrac{397}{1440}(f_{2,0} + f_{0,2}) - \tfrac{63}{32} f_{1,1} + \tfrac{11}{224}(f_{4,0} + f_{0,4}) + \\
&\quad + \tfrac{317}{288} f_{2,2} - \tfrac{37}{160}(f_{3,2} + f_{2,3}), \\
\mu_{1,0}(f) &= \tfrac{131}{60} f_{1,0} + \tfrac{9}{40} f_{2,0} - \tfrac{1}{120} f_{4,0} - \tfrac{173}{300} f_{0,1} - \tfrac{13}{40} f_{1,1} - \tfrac{47}{60} f_{2,1} + \\
&\quad + \tfrac{3}{50} f_{3,1} + \tfrac{1}{4} f_{2,2} + \tfrac{3}{20} f_{0,3} - \tfrac{7}{40} f_{1,3}, \\
\mu_{1,1}(f) &= -\tfrac{4}{15} f_{0,0} - \tfrac{2}{15}(f_{2,0} + f_{0,2}) + \tfrac{33}{20} f_{1,1} - \tfrac{1}{20}(f_{2,1} + f_{1,2}) - \\
&\quad - \tfrac{1}{15} f_{2,2} + \tfrac{1}{40}(f_{3,1} + f_{1,3}).
\end{aligned}$$

Along the lower edge, for $i = 2, \ldots, m-1$, we have

$$\begin{aligned}\mu_{i,0}(f) &= \tfrac{12}{5} f_{i,0} - \tfrac{9}{8} f_{i,1} - \tfrac{1}{4}(f_{i-1,1} + f_{i+1,1}) + \\ &\quad + \tfrac{1}{8}(f_{i-1,2} + f_{i+1,2}) - \tfrac{1}{40} f_{i,3}, \\ \mu_{i,1}(f) &= -\tfrac{7}{30} f_{i,0} - \tfrac{1}{12}(f_{i-1,0} + f_{i+1,0}) + \tfrac{13}{8} f_{i,1} - \tfrac{1}{6} f_{i,2} - \\ &\quad - \tfrac{1}{24}(f_{i-1,2} + f_{i+1,2}) + \tfrac{1}{40} f_{i,3},\end{aligned}$$

and analogous formulae along the three other edges of Ω.

3.3 Norm Estimates

Now, we study the infinity norms of both operators, Q_1 and Q_2.

We define the quasi-Lagrange functions $L_\alpha^{(v)}$, $v = 1, 2$, by the following expression of $Q_1 f$ and $Q_2 f$

$$Q_1 f = \sum_{\alpha \in A} \lambda_\alpha(f) B_\alpha = \sum_{\alpha \in A} f_\alpha L_\alpha^{(1)},$$

$$Q_2 f = \sum_{\alpha \in A} \mu_\alpha(f) B_\alpha = \sum_{\alpha \in A} f_\alpha L_\alpha^{(2)}.$$

We know that the infinity norm of such operators is equal to the Chebyshev norm of its Lebesgue function defined by:

$$\Lambda_v = \sum_{\alpha \in A} |L_\alpha^{(v)}|, \quad v = 1, 2.$$

Using a computer algebra system, we have computed the maxima of Λ_v in some square cells near the origin (south-west vertex of Ω), see Fig. 10, for both operators. The approximate values of these maxima are given in Table 1.

Table 1. Maximum value of the Lebesgue functions Λ_v, $v = 1, 2$ near the origin

(i, j)	(0,0)	(1,0)	(2,0)	(1,1)	(2,1)	(2,2)
$\Lambda_1\|_{\Omega_{i,j}}$	2.00	1.96	1.65	1.63	1.50	1.50
$\Lambda_2\|_{\Omega_{i,j}}$	2.83	1.86	1.70	1.81	1.67	1.50

Theorem 1. *For the operator Q_v, $v = 1, 2$ the following bounds are valid*

$$\|Q_v\|_\infty \leq v + 1, \quad v = 1, 2.$$

We remark that if $A = \mathbb{Z}^2$ the two discrete operators Q_1 and Q_2 coincide and their norm is bounded by 1.5 [12].

4 Numerical Results

In this section we present some numerical results obtained by a computational procedure developed in a Matlab environment [8].

We approximate the following functions (Figs. 11÷13)

$$f_1(x,y) = \frac{1}{1+x^2+y^2}, \quad f_2(x,y) = \ln(1+x^2+y^2),$$

on the square $[-1,1]\times[-1,1]$, and the Franke's function (see e.g. [13])

$$\begin{aligned} f_3(x,y) = & \tfrac{3}{4}\exp\left(-\tfrac{1}{4}\left((9x-2)^2+(9y-2)^2\right)\right) + \\ & + \tfrac{3}{4}\exp\left(-\left(\tfrac{(9x+1)^2}{49} - \tfrac{(9y+1)^2}{10}\right)\right) + \\ & + \tfrac{1}{2}\exp\left(-\tfrac{1}{4}\left((9x-7)^2+(9y-3)^2\right)\right) - \\ & - \tfrac{1}{5}\exp\left(-\left((9x-4)^2+(9y-7)^2\right)\right) \end{aligned}$$

on the square $[0,1]\times[0,1]$.

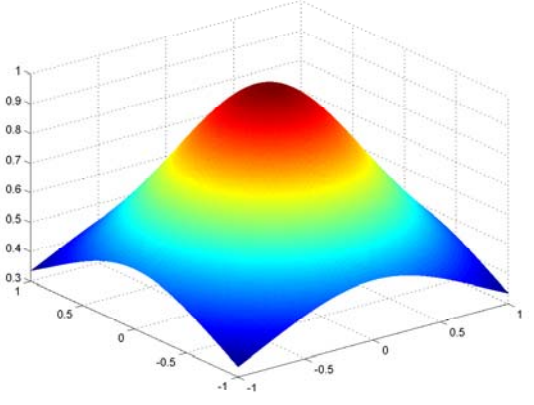


Fig. 11. Function $f_1(x,y)$ on $[-1,1]\times[-1,1]$

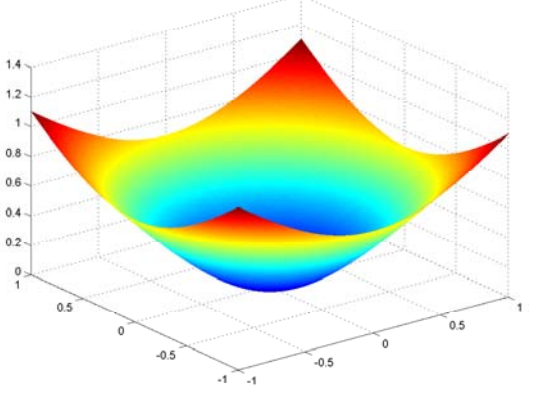


Fig. 12. Function $f_2(x,y)$ on $[-1,1]\times[-1,1]$

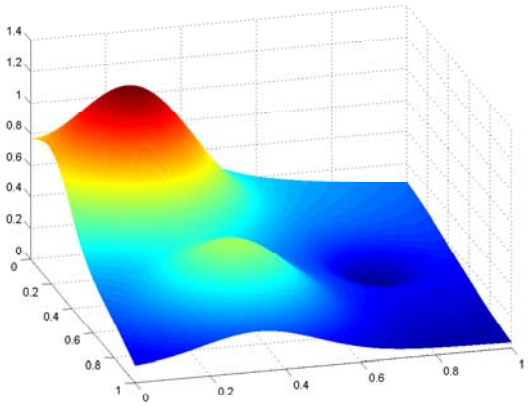


Fig. 13. Function $f_3(x,y)$ on $[0,1]\times[0,1]$

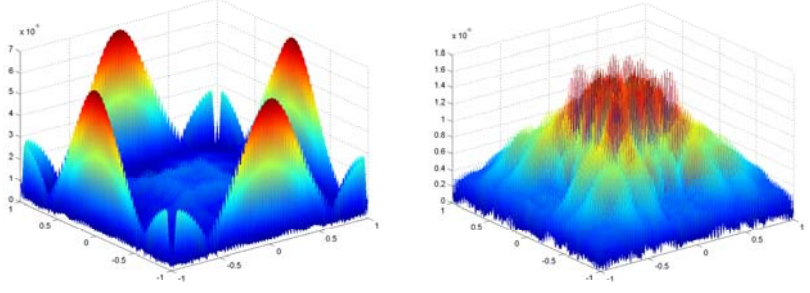


Fig. 14. $|Q_1 f_1 - f_1| \leq 6.7495 \cdot 10^{-6}$ and $|Q_2 f_1 - f_1| \leq 1.9089 \cdot 10^{-6}$ with $m = n = 64$

For each test function, we compute, using a 2000×2000 uniform rectangular grid G of evaluation points in the domain, the maximum absolute error $E_v f = \max_{x\in G} |f(x) - Q_v f(x)|$, $v = 1,2$, for increasing values of m and n, and the logarithm of the ratio between two consecutive errors, $r_v f$, $v = 1,2$, see Table 2. In Figs. 14 ÷ 16 the corresponding errors between f and the two quasi-interpolants $Q_1 f$ and $Q_2 f$ with $m = n = 64$ are shown.

If we evaluate the error at the points where superconvergence holds, Fig. 6, we can observe that with the operator Q_2 the error is $\mathcal{O}(h^4)$, see Table 3.

Although the infinity norm of the operator Q_2 is greater than the infinity norm of Q_1, we can notice that the overall error is smaller.

Furthermore, from Figs. 14 (right) ÷ 16 (right), it seems that the operator Q_2 has a good global error while the operator Q_1 presents a greater error near the boundary as shown in Figs. 14 (left) ÷ 16 (left).

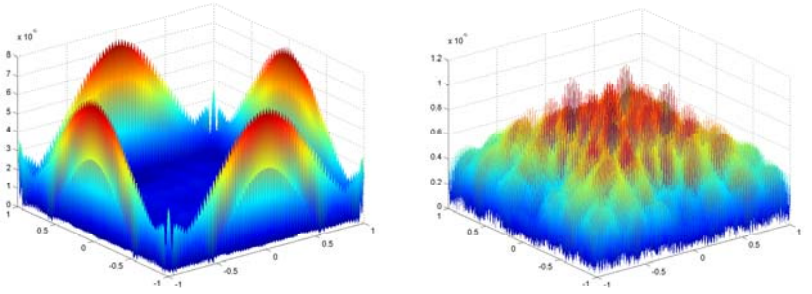


Fig. 15. $|Q_1 f_2 - f_2| \leq 7.5843 \cdot 10^{-6}$ and $|Q_2 f_2 - f_2| \leq 1.1194 \cdot 10^{-6}$ with $m = n = 64$

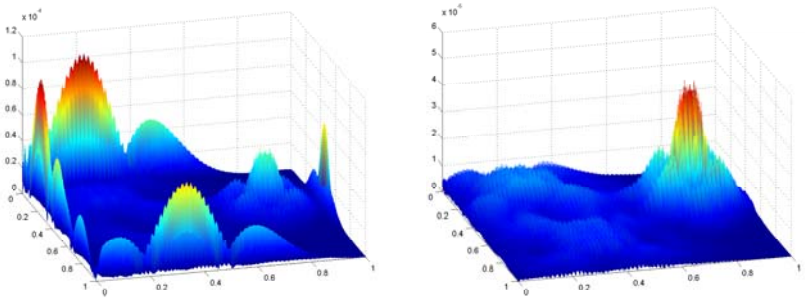


Fig. 16. $|Q_1 f_3 - f_3| \leq 1.0417 \cdot 10^{-4}$ and $|Q_2 f_3 - f_3| \leq 5.0880 \cdot 10^{-5}$ with $m = n = 64$

Table 2. Maximum absolute errors

$m = n$	$E_1 f_1$	$r_1 f_1$	$E_2 f_1$	$r_2 f_1$
32	5.5137(-005)		2.1396(-005)	
64	6.7495(-006)	3.03	1.9089(-006)	3.49
128	8.4039(-007)	3.01	2.0738(-007)	3.20
256	1.0472(-007)	3.00	2.4737(-008)	3.07
512	1.3068(-008)	3.00	3.0335(-009)	3.03
1024	1.6320(-009)	3.00	3.7690(-010)	3.01

$m = n$	$E_1 f_2$	$r_1 f_2$	$E_2 f_2$	$r_2 f_2$
32	6.3773(-005)		1.1627(-005)	
64	7.5843(-006)	3.07	1.1194(-006)	3.38
128	9.2326(-007)	3.04	1.2723(-007)	3.14
256	1.1313(-007)	3.03	1.5380(-008)	3.05
512	1.3957(-008)	3.02	1.8994(-009)	3.02
1024	1.7389(-009)	3.00	2.3611(-010)	3.01

$m = n$	$E_1 f_3$	$r_1 f_3$	$E_2 f_3$	$r_2 f_3$
32	9.1190(-004)		7.2117(-004)	
64	1.0417(-004)	3.13	5.0880(-005)	3.83
128	1.2172(-005)	3.10	4.1378(-006)	3.62
256	1.4577(-006)	3.06	4.1153(-007)	3.33
512	1.7719(-007)	3.04	4.7279(-008)	3.12
1024	2.1932(-008)	3.01	5.7982(-009)	3.03

Table 3. Maximum absolute errors at specific points (Sect. 2.2)

$m=n$	$E_1 f_1$	$r_1 f_1$	$E_2 f_1$	$r_2 f_1$
32	5.3905(-005)		1.8852(-005)	
64	6.7482(-006)	3.00	1.1886(-006)	3.99
128	8.4015(-007)	3.01	7.4451(-008)	4.00
256	1.0470(-007)	3.00	4.6558(-009)	4.00
512	1.3064(-008)	3.00	2.9102(-010)	4.00
1024	1.6314(-009)	3.00	1.8190(-011)	4.00
$m=n$	$E_1 f_2$	$r_1 f_2$	$E_2 f_2$	$r_2 f_2$
32	6.2292(-005)		9.4628(-006)	
64	7.4389(-006)	3.07	5.9488(-007)	3.99
128	9.0720(-007)	3.04	3.7235(-008)	4.00
256	1.1196(-007)	3.02	2.3280(-009)	4.00
512	1.3905(-008)	3.01	1.4551(-010)	4.00
1024	1.7325(-009)	3.00	9.0949(-012)	4.00
$m=n$	$E_1 f_3$	$r_1 f_3$	$E_2 f_3$	$r_2 f_3$
32	8.7514(-004)		7.1035(-004)	
64	1.0120(-004)	3.11	4.7068(-005)	3.92
128	1.1913(-005)	3.09	3.0205(-006)	3.96
256	1.4376(-006)	3.05	1.9001(-007)	3.99
512	1.7638(-007)	3.03	1.1886(-008)	4.00
1024	2.1836(-008)	3.01	7.4321(-010)	4.00

5 Final Remarks

In this paper we have defined and analysed C^1 quadratic discrete quasi-interpolants, constructing their coefficient functionals in several ways, comparing them and giving norm estimates.

In a similar way, we are investigating the construction of good coefficient functionals for C^2 quartic spline quasi-interpolants on criss-cross triangulations and C^2 cubic spline quasi-interpolants on Powell-Sabin triangulations [10].

We plan to use these quasi-interpolants in the second stage of the two-stage method of Schumaker [21].

Furthermore we will apply these operators to the approximation of critical points and curvatures of a surface [12].

Acknowledgement

The author is grateful to Prof. C. Dagnino and Prof. P. Sablonnière for helpful discussions and comments.

References

1. Barrera, D., Ibáñez, M.J., Sablonnière, P.: Near-best discrete quasi-interpolants on uniform and nonuniform partitions. In: Cohen, A., Merrien, J.L., Schumaker, L.L. (eds.) Curve and Surface Fitting, Saint-Malo 2002, pp. 31–40. Nashboro Press, Brentwood (2003)

2. Barrera, D., Ibáñez, M.J., Sablonnière, P., Sbibih, D.: Near-best quasi-interpolants associated with H-splines on a three-direction mesh. J. Comput. Appl. Math. 183, 133–152 (2005)
3. Barrera, D., Ibáñez, M.J.: Bernstein-Bézier representation and near-minimally normed discrete quasi-interpolation operators. Appl. Num. Math. 58, 59–68 (2008)
4. Barrera, D., Ibáñez, M.J., Sablonnière, P., Sbibih, D.: On near best discrete quasi-interpolation on a four-directional mesh. J. Comput. Appl. Math. (to appear)
5. de Boor, C., Höllig, K., Riemenschneider, S.: Box splines. Springer-Verlag, New York (1993)
6. Chui, C.K.: Multivariate splines. SIAM, Philadelphia (1988)
7. Chui, C.K., Wang, R.H.: On a bivariate B-splines basis. Scientia Sinica XXVII, 1129–1142 (1984)
8. Dagnino, C., Lamberti, P.: On C^1 quasi interpolating splines with type-2 triangulations. Progetto MURST Analisi Numerica: Metodi e Software Matematico, Ferrara (2000)
9. Dagnino, C., Sablonnière, P.: Error analysis for quadratic spline QI's on non-uniform criss-cross triangulation of bounded rectangular domains. Quaderni Scientifici del Dipartimento di Matematica, Università di Torino, 38 (2004)
10. Davydov, O., Sablonnière, P.: C^2 cubic spline quasi-interpolants on Powell-Sabin uniform triangulations. In: Buhmann, M.D., Haussmann, W., Jetter, K., Stöckler, J. (eds.) International Conference on Multivariate Approximation, Haus Bommerholz (2008) (submitted)
11. Foucher, F., Sablonnière, P.: Superconvergence properties of some bivariate C^1 quadratic spline quasi-interpolants. In: Cohen, A., Merrien, J.L., Schumaker, L.L. (eds.) Curve and Surface Fitting, Avignon 2006, pp. 160–169. Nashboro Press, Brentwood (2007)
12. Foucher, F., Sablonnière, P.: Approximating partial derivatives of first and second order by quadratic spline quasi-interpolants on uniform meshes. Math. & Computers in Simulation 77, 202–208 (2008)
13. Franke, R.: Scattered data interpolation: Tests of some methods. Math. Comp. 38, 181–200 (1982)
14. Ibáñez Pérez, M.J.: Quasi-interpolantes spline discretos de norma casi mínima. Teoría y aplicaciones. PhD Thesis, Universidad de Granada (2003)
15. Sablonnière, P.: Bases de Bernstein et approximants splines, Thèse de doctorat, Université de Lille (1982)
16. Sablonnière, P.: On some multivariate quadratic spline quasi-interpolants on bounded domains. In: Hausmann, W., et al. (eds.) Modern developments in multivariate approximations. ISNM, vol. 145, pp. 263–278. Birkhäuser Verlag, Basel (2003)
17. Sablonnière, P.: Quadratic spline quasi-interpolants on bounded domains of $\mathbb{R}^d$, $d = 1, 2, 3$. Rend. Sem. Univ. Pol. Torino 61, 61–78 (2003)
18. Sablonnière, P.: Recent progress on univariate and multivariate polynomial and spline quasi-interpolants. In: de Bruijn, M.G., Mache, D.H., Szabados, J. (eds.) Trends and Applications in Constructive Approximation. ISNM, vol. 151, pp. 229–245. Birkhäuser Verlag, Basel (2005)
19. Sablonnière, P.: BB-coefficients of basic quadratic splines on rectangular domains with uniform criss-cross triangulations. Prépublication IRMAR 02-56, Rennes (2002); Corrected version (2007)
20. Sablonnière, P.: Quasi-interpolants splines: exemples et applications. ESAIM: PROC 20, 195–207 (2007)
21. Schumaker, L.L.: Two-stage spline methods for fitting surfaces. In: Schaback, R., Scherer, K. (eds.) Approximation Theory. LNM, vol. 556, pp. 378–389. Springer, Berlin (1976)
22. Watson, G.A.: Approximation Theory and Numerical Methods. Wiley, Chichester (1980)

Sampling and Stability

Christian Rieger[1], Robert Schaback[2], and Barbara Zwicknagl[3]

[1] Institut für numerische Simulation
Hausdorff-Zentrum für Mathematik
Universität Bonn, Wegelerstr. 6
53115 Bonn, Germany
rieger@ins.uni-bonn.de
www.num.math.uni-goettingen.de/crieger
[2] Institut für Numerische und Angewandte Mathematik
Universität Göttingen, Lotzestraße 16-18, D-37083 Göttingen, Germany
schaback@math.uni-goettingen.de
www.num.math.uni-goettingen.de/schaback
[3] Institut für Angewandte Mathematik
Hausdorff-Zentrum für Mathematik
Universität Bonn, Endenicher Allee 60
53115 Bonn, Germany
barbara.zwicknagl@hcm.uni-bonn.de
www.num.math.uni-goettingen.de/zwicknag

Abstract. In Numerical Analysis one often has to conclude that an error function is small everywhere if it is small on a large discrete point set and if there is a bound on a derivative. *Sampling inequalities* put this onto a solid mathematical basis.

A *stability inequality* is similar, but holds only on a finite–dimensional space of trial functions. It allows bounding a trial function by a norm on a sufficiently fine data sample, without any bound on a high derivative.

This survey first describes these two types of inequalities in general and shows how to derive a stability inequality from a sampling inequality plus an inverse inequality on a finite–dimensional trial space. Then the state–of–the–art in sampling inequalities is reviewed, and new extensions involving functions of infinite smoothness and sampling operators using weak data are presented.

Finally, typical applications of sampling and stability inequalities for recovery of functions from scattered weak or strong data are surveyed. These include Support Vector Machines and unsymmetric methods for solving partial differential equations.

1 Introduction

In many practical applications it is necessary to approximate or reconstruct a function as a formula from given strong or weak scattered data. Important examples are domain modeling, surface reconstruction, machine learning or the numerical solution of partial differential equations.

If the strong or weak data used for reconstruction are seen as a *sampling* $S(f) \in \mathbb{R}^N$ of an unknown function f, and if a *trial* function u satisfying $S(u) \approx S(f)$ is calculated

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 347–369, 2010.

in order to recover f, then $S(f-u)$ is small and one has to conclude that $f-u$ is small. Of course, such a conclusion requires additional assumptions, e.g., a bound on derivatives of $f-u$.

The *sampling inequalities* surveyed here quantify the observation that a differentiable function cannot attain large values anywhere if its derivatives are bounded, and if it produces small data on a sufficiently dense discrete set. Along the lines of the above argument, such inequalities are extremely useful in deriving *a priori* error estimates for very general approximation processes [30].

In the univariate setting, sampling inequalities are quite easy to obtain. We assume a sufficiently smooth function f on an interval $[a,b]$ and a discrete ordered set of centers $X = \{x_1,\dots,x_N\} \subset [a,b]$ with

$$a = x_1 < x_2 < \cdots < x_{N-1} < x_N = b$$

and the *fill distance*

$$h_{X,[a,b]} := \sup_{y\in[a,b]} \max_{x\in X} |x-y| = \frac{1}{2} \max_{2\le j\le N} \left|x_j - x_{j-1}\right|, \tag{1}$$

i.e., the largest possible distance any point $x \in [a,b]$ has from the set X. With this notation we consider an arbitrary point $x \in [a,b]$ and its closest point $x_j \in X$ to get

$$\begin{aligned} f(x) &= f(x_j) + \int_{x_j}^{x} f'(t)\,dt, \\ |f(x)| &\le \left|f(x_j)\right| + \sqrt{|x-x_j|}\sqrt{\int_{x_j}^{x} |f'(t)|^2\,dt}, \end{aligned}$$

which yields a first instance of a *sampling inequality*

$$\|f\|_{L_\infty([a,b])} \le \sqrt{h}|f|_{W_2^1[a,b]} + \|S_X(f)\|_{\ell_\infty(\mathbb{R}^N)}$$

for the *sampling operator*

$$S_X \,:\, W_2^1[a,b] \to \mathbb{R}^N \text{ with } S_X(f) := (f(x_1),\dots,f(x_N))^T \quad . \tag{2}$$

This easy example already reveals the basic phenomenon, i.e., it bounds a function in a weak continuous norm in terms of the sampled data on a discrete set X and a strong continuous norm weighted by a power of the fill distance of X. We shall explain this in general in the next section, while we postpone specific applications to sections 3 and 4.

2 Sampling and Stability

Here, we shall exhibit general features of *sampling* and *stability* inequalities and their connections. We admit general spaces of multivariate functions and general sampling operators. Specific cases will follow in Sect. 3.

2.1 Sampling Inequalities

We consider a linear space $\mathscr{F}$ of real–valued functions on some domain $\Omega \subset \mathbb{R}^d$, and assume $\mathscr{F}$ to carry norms $\|\cdot\|_{\mathfrak{S}}$ and $\|\cdot\|_{\mathfrak{W}}$, where $\|\cdot\|_{\mathfrak{S}}$ is stronger than $\|\cdot\|_{\mathfrak{W}}$, i.e.,

$$\|f\|_{\mathfrak{W}} \le C\|f\|_{\mathfrak{S}} \quad \text{for all } f \in \mathscr{F} \ . \tag{3}$$

Here and in the following, C denotes a generic positive constant which is independent of the terms in the following *for all* statements, but sometimes we add dependencies of constants on certain problem parameters by adding argument lists. For example, we could have written $C(\mathscr{F},\mathfrak{S},\mathfrak{W})$ in the inequality (3).

Furthermore, we consider classes $\mathscr{L}$ of finite sets $\Lambda = \{\lambda_1,\dots,\lambda_N\}$ of linearly independent functionals from the dual space $\mathscr{F}^*$ with respect to $\|\cdot\|_{\mathfrak{S}}$. These functionals are used to *sample* a function from $\mathscr{F}$ via the continuous and linear *sampling operator*

$$\begin{aligned} S_\Lambda \ : \ & \mathscr{F} \to \mathbb{R}^N, \\ & f \mapsto (\lambda_1(f),\dots,\lambda_N(f))^T, \end{aligned}$$

generalizing (2). The term h_Λ denotes some discretization parameter which should be small, i.e., $h_\Lambda \overset{N\to\infty}{\longrightarrow} 0$.

Note that the sampling operator contains general linear functionals on rather arbitrary spaces. It allows strong and weak sampling in the sense of point evaluations (*strong sampling*) and distributions that may involve integration (*weak sampling*). Mixing strong and weak sampling is allowed.

Then an abstract form of a sampling inequality is

$$\|f\|_{\mathfrak{W}} \le C\left(h_\Lambda^\sigma \|f\|_{\mathfrak{S}} + C(h_\Lambda)\|S_\Lambda(f)\|_{\mathbb{R}^N}\right) \quad \text{for all } f \in \mathscr{F}, \Lambda \in \mathscr{L} \tag{4}$$

where at this point we do not care which norm on $\mathbb{R}^N$ is chosen. Sometimes such an inequality holds even if $\|\cdot\|_{\mathfrak{S}}$ is replaced by a semi-norm $|\cdot|_{\mathfrak{S}}$ with finite dimensional kernel $\mathscr{P}_{\mathfrak{S}}$, and then we rewrite this as

$$\|f\|_{\mathfrak{W}} \le C\left(h_\Lambda^\sigma |f|_{\mathfrak{S}} + C(h_\Lambda)\|S_\Lambda(f)\|_{\mathbb{R}^N}\right) \quad \text{for all } f \in \mathscr{F}, \Lambda \in \mathscr{L} \ . \tag{5}$$

The exponent $\sigma > 0$ is called *sampling order*. Hence there is a small factor in front of the term with the strong continuous norm and a possibly large term in front of the term with the discrete norm. Furthermore, in most cases the class $\mathscr{L}$ of admissible samplings must be "sufficiently fine" in the sense that $h_\Lambda \le h_0$ holds for some positive constant h_0.

If the sampling operator contains only point evaluations based on a finite point set $X = \{x_1,\dots,x_N\} \subset \Omega$, we write it as

$$S_X := (\delta_{x_1},\dots,\delta_{x_N})^T \tag{6}$$

like in (2), and the discretization parameter is then given by the *fill distance*

$$h_{X,\Omega} := \sup_{y\in\Omega} \max_{x\in X} \|x-y\|_2 \tag{7}$$

of the discrete set X with respect to the domain Ω. Geometrically, the fill distance $h_{X,\Omega}$ can be interpreted as the radius of the largest open ball with center in the closure $\overline{\Omega}$ that

does not contain any of the points from X. It is a useful quantity for the deterministic error analysis in Sobolev spaces, i.e., if there are no further structural assumptions on the approximated functions. On the other hand, the *separation distance* q_X defined by

$$q_X := \frac{1}{2} \min_{j \neq k} \|x_j - x_k\|_2 \tag{8}$$

is the largest radius of balls with centers in X that do not contain any other of the points from X, and it is crucial for stability questions.

If the sampling operators of the class $\mathscr{L}$ consist of evaluations of f and its derivatives on certain finite point sets $X \subset \Omega$, we speak of *strong sampling inequalities*. If some other functionals are involved, which may be well defined even if point evaluation is not continuous, we speak of *weak sampling inequalities*. We shall treat strong and weak sampling separately from Sect. 3 on.

2.2 Special Stability Inequalities

Continuing the notation of (5), we denote the kernel of the semi-norm $|\cdot|_{\mathfrak{S}}$ by $\mathscr{P}_{\mathfrak{S}} \subset \mathscr{F}$. If we insert an element $p \in \mathscr{P}_{\mathfrak{S}}$ from this kernel into the sampling inequality (5), we obtain

$$\|p\|_{\mathfrak{W}} \leq C\left(h_\Lambda^\sigma |p|_{\mathfrak{S}} + C(h_\Lambda) \|S_\Lambda(p)\|_{\mathbb{R}^N}\right) = C(h_\Lambda) \|S_\Lambda(p)\|_{\mathbb{R}^N} \tag{9}$$

for all $p \in \mathscr{P}_{\mathfrak{S}}$ and $\Lambda \in \mathscr{L}$. This means that we can bound a continuous norm by a discrete norm on the data. Bounds of this kind will be called *stability inequalities*. They follow from sampling inequalities and hold on the kernel $\mathscr{P}_{\mathfrak{S}}$ of the strong semi–norm involved, if it is finite-dimensional. If $\mathscr{P}_{\mathfrak{S}}$ is a space of polynomials, and if $S_\Lambda = S_X$ is a strong pointwise sampling operator (6) on a finite set X, these estimates imply Markov-Bernstein inequalities [6]. Let us explain this in some more detail. Assume that $|\cdot|_{\mathfrak{W}} = |\cdot|_{W^1_\infty(\Omega)}$, and that $|\cdot|_{\mathfrak{S}} = |\cdot|_{W^k_\infty(\Omega)}$ for $k > 1$ are classical Sobolev semi-norms. This yields for all $1 \leq \ell \leq d$

$$\begin{aligned} \|\partial_\ell p\|_{L_\infty(\Omega)} &\leq C(h_\Lambda) \|S_X(p)\|_{\ell_\infty(\mathbb{R}^N)} \\ &\leq C(h_\Lambda) \|p\|_{L_\infty(\Omega)} \text{ for all } p \in \pi_{k-1}(\Omega), \end{aligned} \tag{10}$$

where ∂_ℓ denotes the partial derivative in direction of the ℓ-th coordinate. This is a special case of classical Markov-Bernstein-inequalities [6]. Thus it is not surprising that the proofs for sampling inequalities use those classical estimates. The stability inequality (9) implies that the data $S_\Lambda(p)$ contains already enough information about $p \in \mathscr{P}_{\mathfrak{S}}$ to identify p from its discrete data $S_\Lambda(p)$. This is connected to the general concept of *norming sets* introduced by Jetter, Stöckler, and Ward [17]. We shall briefly explain this concept, since it is a direct way for proving stability inequalities under certain circumstances. A finite set Λ of linear functionals is called a *norming set* for $\mathscr{P}_{\mathfrak{S}}$ if the sampling operator

$$\begin{aligned} S_\Lambda|_{\mathscr{P}_{\mathfrak{S}}} : \mathscr{P}_{\mathfrak{S}} &\to S_\Lambda|_{\mathscr{P}_{\mathfrak{S}}}(\mathscr{P}_{\mathfrak{S}}) \subset \mathbb{R}^N \\ v &\mapsto S_\Lambda|_{\mathscr{P}_{\mathfrak{S}}}(v) = (\lambda(v))_{\lambda \in \Lambda} \end{aligned}$$

is injective. Then we can introduce another norm on $\mathscr{P}_{\mathfrak{S}}$ by $\|S_\Lambda|_{\mathscr{P}_{\mathfrak{S}}}(\cdot)\|_{\mathbb{R}^N}$ and we immediately get a stability inequality (9). This explains the terminology *norming set*. We shall explain below why norming sets are crucial in the proofs of sampling inequalities.

2.3 Inverse Inequalities and General Stability Inequalities

Stability inequalities like (9) are limited to the kernel $\mathscr{P}_{\mathfrak{S}}$ of the strong semi-norm in (5). But they should generalize to finite–dimensional trial spaces $\mathscr{R}$ in the sense

$$\|u\|_{\mathfrak{W}} \le C_{stab}(\Lambda, \mathscr{R}, \mathfrak{W}) \|S_\Lambda u\|_{\mathbb{R}^N} \text{ for all } u \in \mathscr{R} \tag{11}$$

bounding a weak continuous norm by a discrete norm. This is obvious due to the fact that all norms on finite–dimensional spaces are equivalent, but it is hard to determine the constants $C(\Lambda, \mathscr{R}, \mathfrak{W})$. Clearly, (11) implies that the sampling operator $S_\Lambda|_{\mathscr{R}} : \mathscr{R} \to \mathbb{R}^N$ is injective, i.e., that Λ is a norming set for $\mathscr{R}$.

To let a sampling inequality produce such a stability inequality, we could use (5) on $\mathscr{R}$ to get

$$\|u\|_{\mathfrak{W}} \le C\left(h_\Lambda^\sigma |u|_{\mathfrak{S}} + C(h_\Lambda) \|S_\Lambda(u)\|_{\mathbb{R}^N}\right) \quad \text{for all } u \in \mathscr{R} \ . \tag{12}$$

The second part of the right–hand side is what we want, but the first part should go away, preferably by moving it to the left–hand side. Thus we want to bound a strong semi-norm by a weaker norm on the trial space like

$$|u|_{\mathfrak{S}} \le C_{inv}(\mathscr{R}, \mathfrak{W}, \mathfrak{S}) \|u\|_{\mathfrak{W}} \text{ for all } u \in \mathscr{R}, \tag{13}$$

and this is called an *inverse inequality*. We point out that such an inequality can hold only if the constant $C_{inv}(\mathscr{R}, \mathfrak{W}, \mathfrak{S})$ grows to infinity if the trial discretization becomes finer and finer. This inequality, however, is independent of the sampling, and together with our sampling inequality (12) it provides the stability inequality (11) with

$$C_{stab}(\Lambda, \mathscr{R}, \mathfrak{W}) = 2CC(h_\Lambda)$$

provided that we have the *stability condition*

$$Ch_\Lambda^\sigma C_{inv}(\mathscr{R}, \mathfrak{W}, \mathfrak{S}) < \frac{1}{2}, \tag{14}$$

which can always be satisfied if h_Λ is sufficiently small, i.e., the sampling is "fine enough". This means that in general

$$\left.\begin{array}{c}\text{Sampling Inequality}\\ +\\ \text{Inverse Inequality}\end{array}\right\} \Rightarrow \text{Stability Inequality.}$$

2.4 Connection to Lebesgue Constants

Sampling and stability inequalities are closely related to Lebesgue constants. To see this, assume a trial space $\mathscr{R}$ that allows unique generalized interpolation for N linearly independent functionals $\lambda_1, \ldots, \lambda_N$ defining a sampling operator S_Λ with values in $\mathbb{R}^N$. Then we can use generalized Lagrange interpolation [39] to build the generalized cardinal interpolants u_{λ_i} from $\mathscr{R}$ with $\lambda_j(u_{\lambda_i}) = \delta_{i,j}$, where $\delta_{i,j}$ denotes the Kronecker

symbol, and the interpolant to a function f is $I_\Lambda(f)(\cdot) = \sum_{j=1}^N \lambda_j(f)\, u_{\lambda_j}(\cdot)$. This yields a stability estimate of the form

$$\begin{aligned}\|I_\Lambda(f)\|_{\mathfrak{W}} &= \left\| \sum_{j=1}^N \lambda_j(f)\, u_{\lambda_j} \right\|_{\mathfrak{W}} \le \sum_{j=1}^N |\lambda_j(f)| \left\| u_{\lambda_j} \right\|_{\mathfrak{W}} \\ &\le \max_{1\le j\le N} |\lambda_j(f)| \sum_{j=1}^N \left\| u_{\lambda_j} \right\|_{\mathfrak{W}} = \|S_\Lambda(f)\|_{\ell_\infty(\mathbb{R}^N)}\, L(\mathscr{R},\Lambda,\mathfrak{W})\end{aligned}$$

for all $f \in \mathscr{F}$, where the *Lebesgue constant* is defined by

$$L(\mathscr{R},\Lambda,\mathfrak{W}) := \sum_{j=1}^N \left\| u_{\lambda_j} \right\|_{\mathfrak{W}} .$$

Now we explain how sampling inequalities can lead to bounds on Lebesgue constants under suitable conditions. If we measure the discrete term in the $\ell_\infty(\mathbb{R}^N)$ norm, i.e.,

$$\|S_\Lambda(u)\|_{\ell_\infty(\mathbb{R}^N)} = \max_{1\le i\le N} |\lambda_i(u)|,$$

we immediately get $\left\|S_\Lambda\left(u_{\lambda_i}\right)\right\|_{\ell_\infty(\mathbb{R}^N)} = 1$ for all $1 \le i \le N$. Applying the sampling inequality (5) yields

$$\begin{aligned}\left\|u_{\lambda_i}\right\|_{\mathfrak{W}} &\le C_1 \left(h_\Lambda^\sigma |u_{\lambda_i}|_{\mathfrak{S}} + C_2(h_\Lambda) \left\|S_\Lambda\left(u_{\lambda_i}\right)\right\|_{\ell_\infty(\mathbb{R}^N)} \right) \\ &\le C_1 \left(h_\Lambda^\sigma |u_{\lambda_i}|_{\mathfrak{S}} + C_2(h_\Lambda) \right) .\end{aligned}$$

Since we have just one sampling operator, we cannot be sure that a stability condition like (14) holds, but in certain situations (see [13] for the case of pointwise interpolation by translates of positive definite kernels) there may be bounds of the form

$$|u_{\lambda_i}|_{\mathfrak{S}} \le C h_\Lambda^{-\sigma} \text{ for all } i,\ 1 \le i \le N$$

and one has $C_2(h_\lambda) = 1$. This leads to boundedness of the norms $\left\|u_{\lambda_i}\right\|_{\mathfrak{W}}$, and consequently the Lebesgue constant is bounded above by $\mathscr{O}(N)$.

3 Variations of Sampling Inequalities

After describing sampling inequalities in general together with their close connection to stability, we now turn to special cases involving spaces of multivariate functions. We review the first sampling inequalities dating back to 2005 to 2007 and give a proof sketch. Then we turn to recent extensions to functions with unlimited smoothness and weak sampling operators.

Throughout this survey we shall deal with a variety of Sobolev spaces defined as in [7] and based on a bounded Lipschitz domain Ω satisfying an interior cone condition.

3.1 Strong Sampling Inequalities: Finite Smoothness

Using the notation of the previous sections in the special case $\|\cdot\|_{\mathfrak{S}} = \|\cdot\|_{W_p^k(\Omega)}$ and $\|\cdot\|_{\mathfrak{W}} = \|\cdot\|_{W_q^m(\Omega)}$, the condition (3) reduces to

$$W_q^k(\Omega) \hookrightarrow W_p^m(\Omega),$$

i.e., by Sobolev's embedding theorem $k \geq m$ and $k - \frac{d}{q} \geq m - \frac{d}{p}$. Since 2005, several *strong sampling inequalities* for functions $u \in W_p^k(\Omega)$ from Sobolev spaces $W_p^k(\Omega)$ with $1 < p < \infty$ and $k > d/p$, or with $p = 1$ and $k \geq d$ on domains $\Omega \subset \mathbb{R}^d$ have been obtained. As a first step in this direction, Narcowich, Ward and Wendland considered the case of functions with scattered zeros [21]. They proved the existence of positive constants C and h_0 such that the inequality

$$|u|_{W_q^m(\Omega)} \leq C h^{k-m-d\left(\frac{1}{p}-\frac{1}{q}\right)_+} |u|_{W_p^k(\Omega)}$$

holds for all functions $u \in W_p^k(\Omega)$ with $k - m > d/p$ and $S_X(u) = 0$ on arbitrary discrete sets X whose fill distance h in the sense of (7) satisfies $h \leq h_0$. The constants C, h_0 may depend on q, m, p, k, Ω, and d, but not on X, h or u. In [38] this result was generalized to functions with arbitrary values on scattered locations:

Theorem 1. *We assume $1 \leq q \leq \infty$, $\alpha \in \mathbb{N}_0^d$, $k \in \mathbb{N}$, and $1 \leq p < \infty$ with $k > |\alpha| + d/p$ if $p > 1$, or with $k \geq |\alpha| + d$ if $p = 1$. Then there are constants C, $h_0 > 0$ such that*

$$\|D^\alpha u\|_{L_q(\Omega)} \leq C \left(h^{k-|\alpha|-d\left(\frac{1}{p}-\frac{1}{q}\right)_+} |u|_{W_p^k(\Omega)} + h^{-|\alpha|} \|S_X u\|_{\ell_\infty(\mathbb{R}^N)} \right)$$

holds for all $u \in W_p^k(\Omega)$ and all discrete sets $X \subset \Omega$ with sampling operators S_X from (6) and fill distance $h := h_{X,\Omega} \leq h_0$.

A similar result was established by Madych [19] in 2006, namely

$$\|u\|_{L_p(\Omega)} \leq C \left(h^k |u|_{W_p^k(\Omega)} + h^{d/p} \|S_X u\|_{\ell_p} \right) \tag{15}$$

for all $u \in W_p^k(\Omega)$ and all X with $h_{X,\Omega} < h_0$. Arcangéli et al. [2] generalized these sampling inequalities by greatly extending the range of parameters:

Theorem 2. *[2, Thm. 4.1] Let p, q, $\kappa \in [1, \infty]$, and let $r \in \mathbb{R}$ such that $r \geq n$ if $p = 1$, $r > n/p$ if $1 < p < \infty$, or $r \in \mathcal{M}^*$ if $p = \infty$. Likewise, let $i_0 = r - n\left(\frac{1}{p} - \frac{1}{q}\right)_+$ and $\gamma = \max\{p, q, \kappa\}$. Then, there exist two positive constants h_0 and C satisfying the following property: for any finite set $X \subset \bar{\Omega}$ (or $X \subset \Omega$ if $p = 1$ and $n = r$) with fill distance $h := h_{X,\Omega} \leq h_0$, for any $u \in W_p^r(\Omega)$ and for any $l = 0, \ldots, \lceil l_o \rceil - 1$, we have*

$$|u|_{W_q^l(\Omega)} \leq C \left(h^{r-l-n\left(\frac{1}{p}-\frac{1}{q}\right)_+} |u|_{W_p^r(\Omega)} + h^{\frac{n}{\gamma}-l} \|u|_X\|_{\ell_\infty(X)} \right) .$$

If $r \in \mathbb{N}^*$ this bound also holds with $l = l_0$ if either $p < q < \infty$ and $l_0 \in \mathbb{N}$, or $(p, q) = (1, \infty)$, or $p \geq q$.

There are several variations, extensions and applications of such sampling inequalities, including derivative data, inequalities on unbounded domains and applications to spline interpolation and smoothing, see e.g., [2, 3, 10, 19, 38]. In all cases the sampling order depends only on the smoothness difference of the two continuous (semi–)norms involved.

3.2 Proof Sketch

A standard way to prove such sampling inequalities follows the lines of [21] and [38]. For some domain $\mathscr{D}$ star-shaped with respect to a ball, let $\left\{a_j^{(\alpha)} : j = 1,\dots,N\right\}$ be a polynomial reproduction of degree k with respect to a discrete set $X = \{x_1,\dots,x_N\} \subset \mathscr{D}$, i.e.,

$$D^\alpha q(x) = \sum_{j=1}^{N} a_j^{(\alpha)}(x)\, q(x_j)$$

holds for every $\alpha \in \mathbb{N}_0^d$ with $|\alpha| \le k$, all $x \in \mathscr{D}$ and all $q \in \pi_k^d(\mathscr{D})$ where π_k^d denotes the space of all d-variate polynomials of degree not exceeding k. Then we have

$$\begin{aligned}
&|D^\alpha u(x)| \\
&\le |D^\alpha u(x) - D^\alpha p(x)| + |D^\alpha p(x)| \\
&\le \|D^\alpha u - D^\alpha p\|_{L_\infty(\mathscr{D})} + \textstyle\sum_{j=1}^N \left|a_j^{(\alpha)}(x)\right| |p(x_j)| \\
&\le \|D^\alpha u - D^\alpha p\|_{L_\infty(\mathscr{D})} + \textstyle\sum_{j=1}^N \left|a_j^{(\alpha)}(x)\right| \|S_X(p)\|_{\ell_\infty(\mathbb{R}^N)} \\
&\le \|D^\alpha u - D^\alpha p\|_{L_\infty(\mathscr{D})} + \\
&\quad + \textstyle\sum_{j=1}^N \left|a_j^{(\alpha)}(x)\right| \left(\|u-p\|_{L_\infty(\mathscr{D})} + \|S_X u\|_{\ell_\infty(\mathbb{R}^N)}\right)
\end{aligned}$$

for arbitrary $u \in W_p^k(\mathscr{D})$ and any polynomial $p \in \pi_k^d(\mathscr{D})$. Using a polynomial reproduction argument based on *norming sets* [18], the Lebesgue constant can be bounded by $\sum_{j=1}^N \left|a_j^{(\alpha)}\right| \le 2$, if some moderate *oversampling* is allowed which is controlled via a Markov inequality. As a polynomial approximation we choose the averaged Taylor polynomial of degree k ([7, Sect. 2]). This leads to a sampling inequality of the form

$$\begin{aligned}
\|D^\alpha u\|_{L_\infty(\mathscr{D})} \le \frac{C}{(k-|\alpha|)!}\delta_{\mathscr{D}}^{k-d/p}\left(\delta_{\mathscr{D}}^{-|\alpha|} + h^{-|\alpha|}\right)|u|_{W_p^k(\mathscr{D})} \\
+2h^{-|\alpha|}\|S_X u\|_{\ell_\infty(\mathbb{R}^N)}
\end{aligned}$$

where $\delta_{\mathscr{D}}$ denotes the diameter of $\mathscr{D}$. To derive sampling inequalities on a Lipschitz domain Ω satisfying an interior cone condition, we cover Ω by domains $\mathscr{D}$ which are star-shaped with respect to a ball, satisfying $\delta_{\mathscr{D}} \approx h$ (see Duchon [14] for details on such coverings). Global estimates are obtained by summation or maximization over the local estimates.

3.3 Strong Sampling Inequalities: Infinite Smoothness

We now consider strong sampling inequalities for infinitely smooth functions in the sense of [26] where the sampling orders turn out to vary exponentially with the fill distance h. When applied to errors of discretization processes involving analytic functions, such sampling inequalities yield convergence results of exponential order, like those of spectral methods.

Following the above proof sketch, these exponential orders are achieved by appropriately coupling the parameter k, controlling the order of smoothness and the order of polynomial reproduction, to the fill distance h. The key point for relations between k and h is the existence of polynomial reproductions of order k on samplings of fill distance $\mathcal{O}(h)$. Following [26], we handle infinitely smooth functions on a domain Ω by normed linear function spaces $\mathcal{H}(\Omega)$ that can for some fixed $1 \le p < \infty$ be continuously embedded into every classical Sobolev space $W_p^k(\Omega)$. More precisely, for some $p \in [1,\infty)$ and all $k \in \mathbb{N}$ we assume that there are embedding operators $I_k^{(p)}$ and constants $E(k)$ such that

$$I_k^{(p)} : \mathcal{H}(\Omega) \to W_p^k(\Omega) \quad \text{with}$$
$$\left\| I_k^{(p)} \right\|_{\{\mathcal{H} \to W_p^k(\Omega)\}} \le E(k) \quad \text{for all } k \in \mathbb{N}_0 \ .$$

There are various examples of spaces with this property, e.g., Sobolev spaces of infinite order as they occur in the study of partial differential equations of infinite order [1], or reproducing kernel Hilbert spaces of Gaussians and inverse multiquadrics.

In the case of infinitely smooth functions, the sampling order is mainly determined by the asymptotic behaviour of the embedding constants $E(k)$ for $k \to \infty$. Typical examples are $E(k) = 1$ for $W_p^\infty(\Omega)$, or $E(k) \le C^k k^{k/2}$ for the reproducing kernel Hilbert space of Gaussians [26]. A typical result is the following theorem [26]:

Theorem 3. *Suppose that $\Omega \subset \mathbb{R}^d$ is bounded, has a Lipschitz boundary, and satisfies an interior cone condition. If there are constants $0 < \varepsilon \le 1$ and $C_E > 0$ such that the embedding constants are bounded by $E(k) \le C_E^k k^{(1-\varepsilon)k}$ for all $k \in \mathbb{N}$, then for all $1 \le q \le \infty$ there are constants C, C_1 and $h_0 > 0$ such that for all data sets $X \subset \Omega$ with fill distance $h \le h_0$ and sampling operators S_X from (6) the inequality*

$$\|D^\alpha u\|_{L_q(\Omega)} \le e^{C\log(h)/\sqrt{h}} \|u\|_{\mathcal{H}(\Omega)} + C_1 h^{-|\alpha|} \|S_X u\|_{\ell_\infty(\mathbb{R}^N)}$$

holds for all $u \in \mathcal{H}(\Omega)$. Here, the constant C does not depend on the space dimension d.

Similar results for different classes of infinitely smooth functions are obtained in [41].

3.4 Weak Sampling Inequalities

Now we focus on *weak* sampling operators. We consider a set of arbitrary functionals

$$\Lambda = \{\lambda_1, \ldots, \lambda_N\} \subset \left(W_2^k(\Omega)\right)^* \ .$$

These functionals define a *weak sampling operator*

$$S_\Lambda := (\lambda_1, \ldots, \lambda_N)^T$$

and we expect a sampling inequality of the form

$$\|u\|_{L_2(\Omega)} \le C h_\Lambda^k \, |u|_{W_2^k(\Omega)} + C \, \|S_\Lambda u\|_{\ell_\infty(\mathbb{R}^N)} \quad .$$

Such an estimate can hold true only if the functionals Λ are a norming set [18] for the polynomials of degree less than k (see Sect. 2.2). We will present two examples of such functionals which are of current research interest.

Weak Convolution-Type Data. Like in [28], we shall use *convolution-type data* of the form

$$\lambda_j(u) = \int_\Omega K(x - x_j)\, u(x)\, dx \tag{16}$$

to build a sampling operator for weak data. Here $X = \{x_1, \ldots, x_N\} \subset \Omega$ is a discrete set of points, and $K : \mathbb{R}^d \to \mathbb{R}$ is called a *test kernel*. We shall consider only convolution-type data generated by translation invariant test kernels. These weak data are closely related to *Finite Volume Methods*. In the usual Finite Volume Method, one simply chooses $K(\cdot - x_j) \equiv 1$ on supp $K(\cdot - x_j)$. There is a theoretical consideration of this case in [36]. In [28], this situation is generalized to non-stationary convolution type data of the form (16), where the kernel K is of some fixed scale for all translates. In [24], the case of *stationary data* is considered, where the support of the test kernel $K(\cdot - \cdot)$ is scaled with the mesh-norm of X in Ω. This implies that $\lambda_j(f)$ is some weighted local mean of f. To be precise we shall impose the following conditions on K:

Definition 4. *A kernel $K : \mathbb{R}^d \to \mathbb{R}$ is called a stationary test kernel if it satisfies*

1. $\int_\Omega K(x - x_j)\, dx = 1$ *for all* $x_j \in X$,
2. $supp(K(\cdot - x_j)) =: V_j \subset \Omega$,
3. $c_1 h_{X,\Omega} \le diam(V_j) \le c_2 h_{X,\Omega}$, *and*
4. $\left\| K(\cdot - x_j) \right\|_{L_p(\Omega)} \le C h_{X,\Omega}^{-d/q}, \quad \frac{1}{p} + \frac{1}{q} = 1$.

Under these conditions there is the following theorem from [24].

Theorem 5. *For weak sampling functionals (16) using a stationary test kernel, there is a sampling inequality*

$$\|u\|_{L_2(\Omega)} \le C \left(h_{X,\Omega}^k \, \|u\|_{W_2^k(\Omega)} + \|S_\Lambda(u)\|_{\ell_\infty(\mathbb{R}^N)} \right) \tag{17}$$

for all $u \in W_2^k(\Omega)$, $k \ge 0$ and all sets X of centers with sufficiently small fill distance $h_{X,\Omega}$.

Such a sampling inequality will be very useful in the error analysis of numerical methods for weak solutions of partial differential equations, since it yields error estimates for functions in $W_2^1(\Omega)$. These estimates are not covered by the well–established theory since $W_2^1(\Omega)$ is not continuously embedded in $C(\Omega)$ for $\Omega \subset \mathbb{R}^d$ with $d \ge 2$.

Galerkin Methods. Another kind of weak data, which arises naturally in the study of partial differential equations, is *Galerkin data*. Namely, we consider a partial differential equation in its weak formulation

$$\text{find } u \in H \ : \ a(u,v) = F(v) \quad \text{for all } v \in X, \tag{18}$$

where $H = W_2^\tau(\Omega)$ is typically a Sobolev space and $a : H \times H \to \mathbb{R}$ is a bilinear form while $F \in H^*$ is a linear functional. Usually, the solution to (18) lies actually in a subspace $W_2^\sigma(\Omega) \subset \mathfrak{W} = W_2^\tau(\Omega)$ of order $\sigma > \tau$. This difference in smoothness leads to some sampling order. To solve the problem (18) approximatively we use a Ritz-Galerkin approach [7, (2.5.7)] and consider the finite dimensional problem

$$\text{find } u \in V_{K,X} \ : \ a(u,v) = F(v) \ \text{ for all } v \in V_{K,X},$$

where

$$V_{K,X} := \text{span } \left\{ K(\cdot - x_j) \ : \ x_j \in X \right\}$$

with a translation invariant kernel K. To simplify the constants in our sampling inequality, it is convenient to consider an orthonormal basis $\{\phi_j\}_{j=1,\ldots,N}$ of $V_{K,X}$ with respect to the bilinear form $a(\cdot,\cdot)$. We define a sampling operator $S_\Lambda := (\lambda_1, \ldots, \lambda_N)^T$ by

$$\lambda_j(u) := a(u, \phi_j) \quad .$$

If there the unique solution u^* to (18) satisfies $u^* \in W_2^k$ (see, e.g., [7] for a formulation of this condition in terms of a and F), then there is a sampling inequality of the form (12) with $\sigma = k - \tau$ (see [24, Thm. 8.3.1] for details).

4 Sampling and Stability in Reconstruction Processes

The previous sections dealt with sampling and stability inequalities in general and in particular, but now we turn to *applications* of both. These can be described as general *reconstruction problems*, where one tries to recover a function f from a weak or strong data provided by a sampling $S_\Lambda(f)$.

4.1 Testing Trial Functions

Starting from a set

$$\Lambda_E = \{\lambda_1, \ldots, \lambda_{N_E}\} \subset \mathscr{F}^*$$

of linear functionals, we define a *sampling operator* via

$$S_{\Lambda_E} := (\lambda_1, \ldots, \lambda_{N_E})^T \quad .$$

Then we consider given data $S_{\Lambda_E}(f)$ generated by application of the sampling operator S_{Λ_E} to an unknown function f, and we *test* certain *trial functions* u_r to see if they reproduce the data, i.e., if

$$S_{\Lambda_E}(u_r) = S_{\Lambda_E}(f) \tag{19}$$

holds. Thus *sampling* here is the same as *testing*. This viewpoint is borrowed from Petrov–Galerkin methods, where an approximate solution is taken from a space of *trial functions*, and it has to satisfy a number of *test equations*. There is a lot of literature in this field which cannot be covered here but the selection [4, 5, 23, 40] should give an overview. In the Finite Element Method, these test equations are *weak*, since they are integrals against test functions, while in collocation methods the test equations are *strong*, i.e., they are evaluations of functions or their derivatives at discrete points. Evaluating test data on trial functions is nothing else than *sampling* them in the sense of this survey.

Consequently, we shall carefully distinguish between the *test* and the *trial* side. The *test* side consists of the sampling operator S_{Λ_E} based on the given functionals Λ_E. The *trial* side consists of a finite dimensional *trial space* $\mathscr{R}$ which is used to generate an approximate solution $u_r \in \mathscr{R}$ to the problem. Typical examples for trial spaces are linear hulls of translates of a kernel or finite element spaces. Note that we use letters like $\mathscr{R}$, r etc. for the t**R**ial side and E, ε etc. for the t**E**st side. In applications to PDE solving, sampling and testing involves a large variety of functionals. For instance, in strong collocation methods for a Poisson problem

$$\begin{aligned} -\Delta u &= g \quad \text{in } \Omega \\ u &= \varphi \quad \text{on } \partial\Omega \end{aligned}$$

there will be functionals of the forms

$$\begin{aligned} \lambda_j(u_r) &:= -\Delta u_r(x_j),\ x_j \text{ in } \Omega, \quad 1 \le j \le N_\Omega \\ \lambda_j(u_r) &:= \quad u_r(x_j),\ x_j \text{ on } \partial\Omega,\ N_\Omega < j \le N_E, \end{aligned}$$

and we assume that there is some f that provides sampled data $\lambda_j(f)$ to yield the test equations

$$\begin{aligned} \lambda_j(u_r) &= -\Delta u_r(x_j) = -\Delta f(x_j) = g(x_j),\ 1 \le j \le N_\Omega \\ \lambda_j(u_r) &= \quad u_r(x_j) = \quad f(x_j) = \varphi(x_j),\ N_\Omega < j \le N_E \end{aligned}$$

which are at least approximately solved for a trial function u_r. Weak methods will replace point evaluations by local integrals against test functions, but our common viewpoint is that trial functions u_r are tested by sampling them and comparing the sampled data $\lambda_j(u_r)$ to the given sampled data $\lambda_j(f)$ of an unknown exact solution f.

Error estimates will take the form of upper bounds of $\|u_r - f\|$ in some norm. But if $dist(f, \mathscr{R})$ is large, there is no testing strategy that can lead to small errors. Thus users must care first for a trial space $\mathscr{R}$ which allows good approximations u_r to the function f providing the test data. This fact is often overlooked, and it is independent of how testing is done. The job of testing is to enable the user to select one good approximation $u_r \in \mathscr{R}$ to f in an efficient and stable way. Thus the *error* of the reconstruction process depends crucially on the *trial* side, while the *test* side cares for the *stability* of the reconstruction.

When testing trial functions, *interpolation* requires the trial functions to reproduce the test data exactly, whereas *approximation* allows small errors in data reproduction. While in some applications interpolation is required, others, in particular those involving errors or noise in the given data, prefer approximation methods. Sometimes certain

parameters are used to control the error in data reproduction, e.g., in machine learning or spline smoothing, and these parameters are closely related to regularization. Examples will follow later, after we have described our favourite trial spaces.

4.2 Kernel–Based Trial Spaces

Throughout the rest of this survey, we shall mainly restrict ourselves to kernel-based methods, i.e., to "meshless" trial spaces spanned by translates of kernels. In their simplest form, they define a trial space $\mathscr{R}$ generated by translates of a single *trial kernel* $K(\cdot,\cdot)$ via

$$\mathscr{R} := \operatorname{span}\left\{K(x_j,\cdot) \ : x_j \in X_R\right\},$$

where the set $X_R := \{x_1,\ldots,x_{N_R}\} \subset \Omega$ is called the set of *trial points*. To avoid complications, our trial kernels $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ will always be symmetric, continuous and positive definite on $\mathbb{R}^d$, though we shall often restrict them to a domain $\Omega \subseteq \mathbb{R}^d$. They are reproducing kernels in a *native* Hilbert space [37] $\mathscr{N}_K(\Omega)$ of functions on $\Omega \subseteq \mathbb{R}^d$ in the sense

$$f(x) = (f, K(x,\cdot))_{\mathscr{N}_K} \text{ for all } f \in \mathscr{N}_K(\Omega),\ x \in \Omega \subseteq \mathbb{R}^d,$$

and $\mathscr{N}_K(\Omega)$ is continuously embedded in $C(\Omega)$. Typical examples are provided by Gaussians $K(x,y) = \exp\left(-\|x-y\|_2^2\right)$ or Sobolev–Matérn kernels

$$K_{\tau-d/2}\left(\|x-y\|_2\right)\|x-y\|_2^{\tau-d/2}$$

for $\tau > d/2$ with the Bessel functions K_ν of second kind. In the latter case one has $\mathscr{N}_K(\Omega) = W_2^\tau(\Omega)$, and certain compactly supported *Wendland kernels* are reproducing in Hilbert spaces which are norm–equivalent to Sobolev spaces. See [37] for details. To arrive at useful sampling orders, we shall assume a continuous embedding

$$\mathscr{N}_K(\Omega) \subseteq W_2^\tau(\Omega) \tag{20}$$

for some $\tau > 0$. Concerning the dependence of native Hilbert spaces $\mathscr{N}_K(\Omega)$ on the domain Ω and related extension/restriction mappings, we again have to refer to [37] for details. For an overview on the use of kernels, in particular in partial differential equations, see the recent review [30] and the references therein.

By picking trial kernels of limited smoothness, we confine ourselves in this section to the case of finitely smooth trial spaces. This is just for the sake of simplicity. Most of the issues can be carried over to the infinitely smooth case using analytic kernels like the Gaussian.

Kernel–based trial spaces allow to align the trial side formally with the test side. This is done by a second set

$$M_R = \{\mu_1,\ldots,\mu_{N_R}\} \subset \mathscr{N}_K^*(\Omega)$$

of linear functionals and a second *sampling operator* for trial purposes via

$$S_{M_R} := (\mu_1,\ldots,\mu_{N_R})^T$$

to define a generalized kernel–based trial space

$$\mathscr{R}_{M_R} := \operatorname{span}\left\{\mu_j^x K(x,\cdot) \ : \mu_j \in M_R\right\} \subset \mathscr{N}_K(\Omega) \tag{21}$$

where we denote the action of μ_j on the variable x by μ_j^x. The two sampling operators S_{M_R} and S_{Λ_E} will allow us to use sampling inequalities on both the trial and the test side.

Introducing coefficient vectors $\alpha = (\alpha_1, \ldots, \alpha_{N_R})^T \in \mathbb{R}^{N_R}$ we can write the trial functions from $\mathscr{R}_{M_R}$ as

$$u_\alpha(\cdot) := \sum_{j=1}^{N_R} \alpha_j \mu_j^x K(x,\cdot) = \alpha^T S_{M_R}^x K(x,\cdot) \ . \tag{22}$$

4.3 Symmetric Interpolation Methods

In *symmetric methods* the test and trial sides are the same, i.e., we set $S_{M_R} = S_{\Lambda_E}$ and $N_R = N_E$, allowing to skip the R and E indices for this subsection and to use $\Lambda = \{\lambda_1, \ldots, \lambda_N\} \subset \mathscr{N}_K(\Omega)^*$ throughout. Each trial function will be of the form (22), and the interpolation system (19) takes the form of a linear equation for the coefficient vector α

$$S_\Lambda(u_\alpha) = A_{\Lambda,\Lambda}\,\alpha = S_\Lambda(f), \tag{23}$$

where the *kernel* matrix

$$A_{\Lambda,\Lambda} := \left(\lambda_i^x \lambda_j^y K(x,y)\right)_{1 \le i,j \le N} \tag{24}$$

is symmetric and positive definite if the functionals in $\Lambda \subset \mathscr{N}_K(\Omega)^*$ are linearly independent and if the kernel K is positive definite. Under these assumptions, the above system is uniquely solvable [39] for data given by arbitrary functions $f \in \mathscr{N}_K(\Omega)$.

We now sketch how sampling inequalities can be used for a fully general error analysis of symmetric interpolation. We point out that sampling inequalities can also be used for non-interpolatory and unsymmetric recovery processes, as will be shown in later sections.

Assume that the sampling operator S_Λ allows a sampling inequality of the form (5) on a space $\mathscr{F}$ into which $\mathscr{N}_K(\Omega)$ can be continuously embedded. Then we have

$$\|f - u^*(f)\|_{\mathfrak{W}} \le C h_\Lambda^\sigma \, |f - u^*(f)|_{\mathscr{F}} \quad \text{for all } f \in \mathscr{N}_K(\Omega)$$

if we denote the solution of the interpolation system by $u^*(f)$. Using the embedding $\mathscr{N}_K(\Omega) \subset \mathfrak{S}$ and the standard minimum–norm property [37,39]

$$\|u^*(f)\|_{\mathscr{N}_K(\Omega)} \le \|f\|_{\mathscr{N}_K(\Omega)}$$

of solutions of systems like (23) this implies the error bound

$$\|f - u^*(f)\|_{\mathfrak{W}} \le C h_\Lambda^\sigma \, \|f\|_{\mathscr{N}_K(\Omega)} \quad \text{for all } f \in \mathscr{N}_K \ .$$

We shall give a few examples.

Symmetric Strong Interpolation. The simplest case takes a finite subset X of Ω with fill distance (7) and assumes (20). Then we can use the above argument to apply Theorem 1 and get an error bound

$$\|f - u^*(f)\|_{L_q(\Omega)} \le Ch^{\tau - d\left(\frac{1}{2} - \frac{1}{q}\right)_+} \|f\|_{\mathcal{N}_K(\Omega)} \tag{25}$$

for all $f \in \mathcal{N}_K$ and all sets X with sufficiently small fill distance h. This reproduces the well known error bounds (see [37]). If some other functionals are added, such a bound will still hold.

Symmetric Weak Interpolation. In the case of weak convolution-type data based on a set $X = \{x_1, \ldots, x_N\}$ as defined in (16), i.e.,

$$\lambda_j(u) = \int_\Omega \mathfrak{E}(x - x_j)\, u(x)\, dx \quad j = 1, \ldots, N$$

with a stationary test kernel $\mathfrak{E}$ as in Definition 4) the kernel matrix $A_{\Lambda,\Lambda}$ as defined in (24) has entries

$$\left(A_{\Lambda,\Lambda}\right)_{i,j} := \int_\Omega \int_\Omega \mathfrak{E}(x - x_i)\, \mathfrak{E}(y - x_j)\, K(x,y)\, dxdy$$

where the kernel K is symmetric, positive definite and satisfies (20). Then the sampling inequality of the form (17) yields as a direct consequence

Theorem 6. *Under the above assumptions, recovery of functions f by symmetric weak interpolation leads to interpolants $u^*(f)$ with error bounds*

$$\|f - u^*(f)\|_{L_2(\Omega)} \le Ch_{X,\Omega}^{\tau} \|f\|_{\mathcal{N}_K(\Omega)} \tag{26}$$

for all $f \in \mathcal{N}_K(\Omega)$ and all discrete sets X with sufficiently small fill distance $h_{X,\Omega}$.

4.4 Symmetric Non-interpolatory Methods

Although interpolation processes lead to good approximation properties, they have some drawbacks, e.g., the condition of the system (23) is dominated by the separation distance q_X of (8) which can be considerably smaller than the fill distance. In particular, if an additional point comes close to a point of X, the system condition deteriorates dramatically, but omission of the new point would do no harm and can be viewed as a regularization of the augmented system. This indicates that regularized methods are important even in the case of noise-free data. We shall deal with these now.

Approximation as Regularized Interpolation. Here, we outline how sampling inequalities can be used to derive worst-case convergence rates for regularized reconstruction processes. We shall concentrate on regularization methods that avoid exact solving of the system (23).

Besides improving condition numbers, most regularization methods have several advantages, e.g., regularization is closely related to *sparse approximation* [15]. The crucial point for the analysis of all regularized reconstruction processes Π^ν, where ν is a regularization parameter, is to show the following two properties

$$\|\Pi^\nu(f)\|_{W_2^\tau(\Omega)} \le \|f\|_{W_2^\tau(\Omega)} \quad \text{and}$$
$$\max_{1\le j\le N} \left|\lambda_j(f-\Pi^\nu f)\right| \le g(\nu,f)\,\|f\|_{W_2^\tau(\Omega)}$$

where the function $g(\nu,f)$ determines the approximation quality of Π^ν. These properties can be seen as *stability* and *consistency*. We give two examples where these properties are successfully coupled with sampling inequalities.

Spline Smoothing. We shall focus on the well-known case of *spline smoothing* or ℓ_2-*spline-regression*. A more detailed overview can be found in [35] and [37]. For a given $f \in W_2^\tau(\Omega)$ and the various functionals λ_j from the previous sections we can formulate the smoothed optimal recovery problem

$$\min_{u\in W_2^\tau(\Omega)} \left(\sum_{j=1}^N \left|\lambda_j(u-f)\right|^2 + \nu\,\|u\|^2_{W_2^\tau(\Omega)} \right), \tag{27}$$

where $\nu \ge 0$ is called the *smoothing parameter*. The case of strong data $\lambda_j = \delta_{x_j}$ is discussed in [38], whereas the case of weak convolution-type data is dealt with in [24]. For a detailed discussion of the smoothing parameter see [35]. We simply note that the special case $\nu = 0$ corresponds to finding a generalized interpolant, i.e., a function $u^{(0)}(f) \in W_2^\tau(\Omega)$ that satisfies the generalized interpolation conditions (19). It is well known [37] that there always exists a solution $u^{(\nu)}(f)$ to this relaxed interpolation problem (27) in the linear space (21). The coefficients $\alpha \in \mathbb{R}^N$ with respect to the basis $\{\lambda_j^x K(\cdot,x)\}$ can be found by solving the linear system

$$\left(A_{\Lambda,\Lambda} + \nu \mathrm{Id}\right)\alpha = f_\Lambda,$$

with the kernel matrix $A_{\Lambda,\Lambda}$ from (24) and

$$f_\Lambda = S_\Lambda(f) = (\lambda_1(f),\dots,\lambda_N(f))^T \quad .$$

As elaborated in [38] for strong data, we have the following two inequalities

$$\left\|u^{(\nu)}(f)\right\|_{W_2^\tau(\Omega)} \le \left\|u^{(\nu)}(f)\right\|_{W_2^\tau(\Omega)} \le \|f\|_{W_2^\tau(\Omega)},$$
$$\left\|S_\Lambda\left(f - s^{(\nu)}(f)\right)\right\|_{\ell_\infty(\mathbb{R}^N)} \le \sqrt{\nu}\,\|f\|_{W_2^\tau(\Omega)} \quad .$$

Inserting into the sampling inequality (1) yields the bound

$$\left\|f - s_f^{(\nu)}\right\|_{L_2(\Omega)} \le C\left(h^\tau + \sqrt{\nu}\right)\|f\|_{W_2^\tau(\Omega)} \quad .$$

This inequality suggests an *a priori* choice of the smoothing parameter as $\nu \le h^{2\tau}$, which leads to the optimal order [29] achievable by interpolation, but now using a linear system with a condition independent of the separation distance q_X.

Kernel-Based Learning. There is a close link between the theories of kernel-based approximation, spline smoothing and machine learning. Since there is a broad collection of literature on this topic (see, e.g., [8,9,11,12,22,30,31,33] and the references therein), we shall only briefly sketch the viewpoint of learning theory. In this section, we deal only with *strong* recovery or approximation problems.

In order to recover a continuous function f from a strong sample like $S_X(f) = (f(x_1),\dots,f(x_N))^T$, the most common choice for an approximant in kernel-based approximation theory u_f is an interpolant, i.e., $S_X(u_f) = S_X(f)$. This method obviously makes the best use of the sample $S_X(f) \in \mathbb{R}^N$. But there are also some drawbacks of classical interpolation. On the one hand, the reconstruction is very unstable if we consider $S_X(f)$ to be spoiled by noise. On the other hand, there are also numerical disadvantages, namely the computation of the interpolant may be ill-conditioned. Furthermore, if $u_f(\cdot) = \sum_{j=1}^N \alpha_j K(\cdot,x_j)$ denotes the interpolant, there will be many non-zero coefficients, i.e., this method is non-sparse.

One way out of these problems is provided by learning theory. Here, the reconstruction problem is an example of *supervised regression*, because the real values $S_X(f)$ are generated by an unknown, but fixed function f. Instead of $S_X(f)$ we may consider more generally a vector of possibly disturbed values $y \approx S_X(f)$. One typically relaxes the interpolation condition by using a more general similarity measure, e.g., by using a *loss function*. A typical example is Vapnik's ε-intensive loss function [34]

$$|f(x)-y|_\varepsilon = \begin{cases} 0 & if\,|f(x)-y| \le \varepsilon \\ |f(x)-y|-\varepsilon & if\,|f(x)-y| > \varepsilon \end{cases},$$

which allows deviations up to a positive parameter $\varepsilon > 0$. A popular reconstruction technique called ν–Support–Vector–Regression (ν–SVR, [32]) in a Hilbert space $H(\Omega)$ of functions $\Omega \to \mathbb{R}$ is then obtained as solution to the optimization problem

$$\min_{\substack{u\in H(\Omega)\\ \varepsilon\in\mathbb{R}^+}} \frac{1}{N}\sum_{j=1}^N |u(x_j)-y_j|_\varepsilon + \varepsilon\nu + \lambda\,\|u\|^2_{H(\Omega)} \tag{28}$$

with an *a priori* chosen parameter λ. In this section we shall focus on the ν–SVR, but everything works as well for similar regression techniques such as the ε-SVR [25,26]. The ν–SVR possesses a solution (u_y,ε^*) [20], and if $H(\Omega)$ is equivalent to the native Hilbert space of a symmetric positive definite kernel K, there is a solution of the form $u_y(\cdot) = \sum_{j=1}^N \alpha_j K(\cdot,x_j)$ which can be computed by solving a finite dimensional quadratic optimization problem for the coefficient vector α [31]. As elaborated in [25], sampling inequalities can be used to provide a worst-case error analysis, even in the case of noisy data. Such bounds require no knowledge about the underlying noise model. As outlined above, we assume that the data $y = y_f \in \mathbb{R}^N$ comes from the strong sample $S_X(f)$ from some function $f \in W_2^\tau(\Omega) \cong \mathcal{N}_K(\Omega)$ on a set $X = \{x_1,\dots,x_N\} \subset \Omega$, but we allow the given data to be corrupted by some additive error $r \in \mathbb{R}^N$, i.e.,

$$S_X(f) = y + r \ . \tag{29}$$

If we solve the optimization problem (28) with data $y = S_X(f) - r$, we obtain an approximant u_y to the generating function f. The following theorem provides an upper bound for the approximation error of this process.

Theorem 7. *Suppose some bounded domain $\Omega \subset \mathbb{R}^d$ with Lipschitz boundary, and $\lfloor \tau - 1 \rfloor > d/2$. Then there are constants C, $h_0 > 0$, depending only on Ω, d, q and τ with the following property. For all function $f \in W_2^\tau(\Omega)$, all sets $X = \{x_1, \ldots, x_N\} \subset \Omega$ with fill distance $h \le h_0$ and all errors $r \in \mathbb{R}^N$, a solution (u_y, ε^*) of (28) with data y being related to the samples of f via (29), satisfies for any $\delta > 0$*

$$\|f - u_y\|_{L_\infty(\Omega)} \le C \left(h^{\tau - \frac{d}{2}} \left(\|f\|_{W_2^\tau(\Omega)} + \sqrt{\frac{1}{N\lambda} \sum_{j=1}^N |r_j|_\delta + \frac{\nu\delta}{\lambda} + \|f\|^2_{W_2^\tau(\Omega)}} \right) + \sum_{j=1}^N |r_j|_\delta + \nu N \delta + \varepsilon^* (1 - N\nu) + N\lambda \|f\|^2_{W_2^\tau(\Omega)} + \|r\|_+ \ell_\infty(\mathbb{R}^N) \right).$$

The bound in Theorem 7 involves a positive parameter δ, which we are free to choose. The optimal choice of δ obviously depends on the problem parameters. Again, the error estimate suggests some choice of the problem parameters. If we assume that the error does not exceed the data itself, i.e., $\|r\|_{\ell_\infty(\mathbb{R}^N)} \le \delta^* \le \|f\|_{W_2^\tau(\Omega)}$, we can choose the parameters λ, ν and δ (see [25] for details) such that

$$\|f - u_y\|_{L_\infty(\Omega)} \le C \left(h^{\tau - d/2} \|f\|_{W_2^\tau(\Omega)} + \delta^* \right) .$$

This shows that the solution of the ν–SVR leads to optimal sampling orders [29] in Sobolev spaces with respect to the fill distance h. These optimal rates are also attained by classical interpolation in the native Hilbert space [37]. The ν–SVR, however, allows for much more flexibility and less complicated solutions.

4.5 Unsymmetric Methods

We now go back to Sect. 4.4.1 and deal with unsymmetric recovery methods where the sampling operator S_{Λ_S} used for testing is different from the sampling operator S_{M_R} defining the trial space. We simplify the trial space to let the sampling operator

$$S_{M_R} = \left(\delta_{x_1}, \ldots, \delta_{x_{N_R}} \right)^T$$

consist of point evaluations on a set of points $X_R := \{x_1, \ldots, x_{N_R}\} \subset \Omega$ only, and thus we use the trial space

$$\mathcal{R} := \operatorname{span} \left\{ K(x_j, \cdot) \ : x_j \in X_R \right\} .$$

We want to recover an unknown function $f \in \mathcal{N}_K(\Omega) \subseteq W_2^\tau(\Omega)$ from its given weak or strong test data

$$S_{\Lambda_E}(f) := (\lambda_1(f), \ldots, \lambda_{N_E}(f))^T$$

using trial functions from $\mathscr{R}$. That is, we seek a function $u_r(f) \in \mathscr{R}$ satisfying

1. $S_{\Lambda_E}(u_r(f)) \approx S_{\Lambda_E}(f)$
2. $\|u_r(f) - f\|_{W_2^m(\Omega)} \leq \varepsilon(r, f, \tau, m)$ small, for some $m < \tau$.

Typically, one has $\varepsilon(r, f, \tau, m) = h_R^{\tau-m} \|f\|_{W_2^\tau(\Omega)}$. Consider the interpolation system

$$S_{\Lambda_E}(u_r(f)) = S_{\Lambda_E}(f)$$

as in (19). Unfortunately, this system is unsymmetric and hence we cannot simply solve a linear system as in the case of symmetric methods. But we use the observation from Sect. 2.3 that under the condition (14) the linear system has full rank. Hence we can apply any numerical method solving the usually overdetermined system (19) to some accuracy by *residual minimization*. We denote by $u^*(f)$ the best approximant from $\mathscr{R}$ to f which is an interpolant to the unknown data $R_{M_R}(f)$ and define $u_r(f)$ by the following property

$$\|S_{\Lambda_E}(f) - S_{\Lambda_E}(u_r(f))\|_{\mathbb{R}^{N_E}} = \inf_{u_r \in \mathscr{R}} \|S_{\Lambda_E}(f) - S_{\Lambda_E}(u_r)\|_{\mathbb{R}^{N_E}} \quad .$$

Then we have

$$\begin{aligned} \|S_{\Lambda_E}(f) - S_{\Lambda_E}(u_r(f))\|_{\mathbb{R}^{N_E}} &= \inf_{u_r \in \mathscr{R}} \|S_{\Lambda_E}(f) - S_{\Lambda_E}(u_r)\|_{\mathbb{R}^{N_E}} \\ &\leq \|S_{\Lambda_E}(f) - S_{\Lambda_E}(u^*(f))\|_{\mathbb{R}^{N_E}} \\ &\leq \varepsilon(r, f, \tau, m) \|S_{\Lambda_E}\|, \end{aligned} \tag{30}$$

where $\|S_{\Lambda_E}\|$ denotes the norm of the test sampling operator as a mapping between $W_2^m(\Omega)$ and $\mathbb{R}^{N_E}$. Note that we used a sampling inequality to bound the interpolation error. This gives the first property. For the second we find

$$\|f - u_r(f)\|_{W_2^m(\Omega)} \leq \|f - u^*(f)\|_{W_2^m(\Omega)} + \|u_r(f) - u^*(f)\|_{W_2^m(\Omega)} \quad .$$

In order to bound the second term on the right-hand side, we apply a stability inequality of the form (11) to the function $u_r(f) - u^*(f) \in \mathscr{R}$ and get

$$\begin{aligned} &\|u_r(f) - u^*(f)\|_{W_2^m(\Omega)} \\ &\leq C_{stab}(m, h_E, \mathscr{R}) \|S_{\Lambda_E}(u_r(f)) - S_{\Lambda_E}(u^*(f))\|_{\mathbb{R}^{N_S}} \\ &\leq \varepsilon(r, f, \tau, m) C_{stab}(m, h_E, \mathscr{R}) \|S_{\Lambda_E}\| \quad . \end{aligned}$$

Now everything combines into

$$\|f - u_r(f)\|_{W_2^m(\Omega)} \leq C\varepsilon(r, f, \tau, m)(1 + C_{stab}(m, h_E, \mathscr{R}) \|S_{\Lambda_E}\|)$$

which reduces under the assumption $\varepsilon(r, f, \tau, m) = h_R^{\tau-m} \|f\|_{W_2^\tau(\Omega)}$ to the usual form

$$\|f - u_r(f)\|_{W_2^m(\Omega)} \leq C h_R^{\tau-m} (1 + C_{stab}(m, h_E, \mathscr{R}) \|S_{\Lambda_E}\|) \|f\|_{W_2^\tau(\Omega)} \quad .$$

4.6 Unsymmetric Weak Recovery

The paper [28] deals with unsymmetric recovery of functions $f \in L_2(\Omega)$ from weak data obtained via nonstationary convolution. Under suitable conditions on the trial and test kernels, the numerical solution $u_r(f)$ has an error bound of the form

$$\|f - u_r(f)\|_{W_2^{-2\tau}} \le C h_R^{2\tau} \|f\|_{L_2(\Omega)} \text{ for all } f \in L_2(\Omega)$$

provided that both kernels have at least smoothness τ in the sense of Sect. 4.2. To show how weak stationary sampling can be put into the framework of the previous section, we follow [24] and combine the functionals from Sect. 3.4.1 with the ideas from the last section and from [28] to derive L_2–error estimates. For convenience we briefly recall the definition of *weak stationary convolution-type data*

$$\lambda_j(u) = \int_\Omega \mathfrak{E}(x - x_j)\, u(x)\, dx$$

with a stationary test kernel $\mathfrak{E}$ as characterized in Definition 4. As outlined in Sect. 4.5 we consider the following problem. An unknown function $f \in W_2^\tau(\Omega)$ has to be recovered approximately from its data $(\lambda_1(f), \ldots, \lambda_N(f))^T$. We know that there is a good but unknown approximation $u^*(f)$ from the trial space $\mathscr{R} = \operatorname{span}\{K(x_j, \cdot) : x_j \in X_R\}$ to the function $f \in W_2^\tau(\Omega)$. Under certain conditions [24], there is an error estimate of the form

$$\|f - u^*(f)\|_{L_2(\Omega)} \le h_R^\tau \|f\|_{W_2^\tau(\Omega)}, \tag{31}$$

showing the expected approximation order. The unsymmetric overdetermined system (19) takes the form

$$\lambda_j(f - u_r) = 0 \text{ for all } 1 \le j \le N_E \ .$$

As shown in (30) we can find a function $u_r(f) \in \mathscr{R}$ which solves this system up to some accuracy, i.e.,

$$\left|\lambda_j(f - u_r(f))\right| \le C h_R^\tau \|S_{\Lambda_E}\| \, \|f\|_{W_2^\tau(\Omega)} \text{ for } 1 \le j \le N_E \ .$$

As pointed out in Sect. 4.5 and in (11) we assume an inverse inequality of the form

$$\|u\|_{W_2^\tau(\Omega)} \le C\gamma(X_R) \|u\|_{L_2(\Omega)} \quad \text{for all } u \in \mathscr{R} \ .$$

Unfortunately, the value of $\gamma(X_R)$ is in general not known. There is a result in this direction in [24], namely if the trial kernel K is a radial basis function with algebraic smoothness $\tau > d/2$ and if X_R is separated from the boundary by q_R, then $\gamma(X_R) \approx q_R^{-\tau}$. Here, q_R denotes the separation distance of X_R. However, we can always make sure that the fill distance of the test data is small enough to stabilize the reconstruction, i.e., we shall assume a coupling of the discretizations as in (14),

$$\gamma(X_R)\, h_E^\tau \le \frac{1}{2C} \quad \text{with } C > 1 \ . \tag{32}$$

Combining everything along the lines of [27] yields the following result:

Theorem 8. *We denote by $u_r(f) \in \mathscr{R}$ the approximate solution of the system (19). Then we have an error bound of the form*

$$\|f - u_r(f)\|_{L_2(\Omega)} \leq C\,(1 + \|S_{\Lambda_E}\|)\, h_R^{\tau} \|f\|_{W_2^{\tau}(\Omega)}$$

for sufficiently fine test discretizations.

In contrast to the results of [28], this error analysis does not assume f to be known on a slightly larger domain $\tilde{\Omega}$. Furthermore, the result holds for the L_2-norm, not for negative order Sobolev norms as in [28]. Unfortunately, the norm of the sampling operator shows up in the final estimate. This norm $\|S_{\Lambda_E}\| \sim h_E^{-d/2}$ grows and hence prevents optimal rates in the error estimate. Furthermore one needs to sassume $\tau > d/2$, which excludes fully weak problems. A promising way out seems to be to drop the inadequate ℓ_∞ norm in favor of $h_E^{d/2}\|S_{\Lambda_E} u\|_{\ell_2(\mathbb{R}^N)}$ like in Madych's sampling inequality (15). But this is work in progress.

5 Outlook

Research on general sampling and stability inequalities is just in its beginnings, but since they are of central importance to Numerical Analysis, they deserve a larger audience via this survey. There are at least two obvious directions for future research. On the one hand-side, there are many more possible applications for which sampling inequalities involving different functionals are required. This might include sampling of parametric curves and surfaces. On the other hand, there are some technical issues for improving the practical applicability. In every theorem, there are several generic constants involved for which one would need better bounds. This interacts with the stability estimates and the coupling of trial and test discretizations. In particular, to provide realistic upper bounds on fill distances on the test side, various generic constants need thorough investigation.

Acknowledgement

We would like to thank the referees and the editor for several useful comments.

References

1. Agadzhanov, A.: Functional properties of Sobolev spaces of infinite order. Soviet. Math. Dokl. 38, 88–92 (1989)
2. Arcangéli, R., López de Silanes, M., Torrens, J.: An extension of a bound for functions in Sobolev spaces, with applications to (m,s)-spline interpolation and smoothing. Numer. Math. 107, 181–211 (2007)
3. Arcangéli, R., López de Silanes, M., Torrens, J.: Estimates for functions in Sobolev spaces defined on unbounded domains. To appear in Journal of Approximation Theory (2009), doi:10.1016/j.jat.2008.09.001

4. Atluri, S., Shen, S.: The meshless local Petrov-Galerkin (MLPG) method: A simple and less-costly alternative to the Finite Element and Boundary Element methods. Computer Modeling in Engineering and Sciences 3, 11–51 (2002)
5. Atluri, S., Shen, S.: The meshless local Petrov-Galerkin (MLPG) method. Tech Science Press, Encino (2002)
6. Borwein, P., Erdelyi, T.: Polynomials and Polynomial Inequalities. Springer, New York (1995)
7. Brenner, S.C., Scott, L.R.: The mathematical theory of finite element methods. Texts in Applied Mathematics, vol. 15. Springer, New York (1994)
8. Caponnetto, A., DeVito, E.: Optimal rates for the regularized least-squares algorithm. Foundations of Computational Mathematics 7, 331–368 (2007)
9. Caponnetto, A., DeVito, E.: Learning and approximation by Gaussians on Riemannian manifolds. Advances in Computational Mathematics 29, 291–310 (2008)
10. Corrigan, A., Wallin, J., Wanner, T.: A sampling inequality for fractional order Sobolev seminorms using arbitrary order data. Preprint, available online via arXiv:0801.4097v2
11. Cucker, F., Smale, S.: On the mathematical foundations of Learning. Bulletin of the AMS 39, 1–49 (2001)
12. Cucker, F., Zhou, D.-X.: Learning Theory: An Approximation Theory Viewpoint. Cambridge University Press, Cambridge (2007)
13. De Marchi, S., Schaback, R.: Stability of kernel-based interpolation. To appear in Advances in Computational Mathematics (2008), doi:10.1007/s10444-008-9093-4
14. Duchon, J.: Sur l'erreur d'interpolation des fonctions de plusieurs variables pas les D^m–splines. Rev. Française Automat. Informat. Rech. Opér. Anal. Numer. 12, 325–334 (1978)
15. Girosi, F.: An Equivalence Between Sparse Approximation and Support Vector Machines. Neural Computation 10, 1455–1480 (1998)
16. Haroske, D.D., Triebel, H.: Distributions, Sobolev spaces, elliptic equations. EMS Textbooks in Mathematics. European Mathematical Society (EMS), Zürich (2008)
17. Jetter, K., Stöckler, J., Ward, J.D.: Norming sets and scattered data approximation on spheres. In: Approximation Theory IX. Computational Aspects, vol. II, pp. 137–144 (1998)
18. Jetter, K., Stöckler, J., Ward, J.D.: Error estimates for scattered data interpolation on spheres. Mathematics of Computation 68, 733–747 (1999)
19. Madych, W.R.: An estimate for multivariate interpolation II. Journal of Approximation Theory 142, 116–128 (2006)
20. Micchelli, C.A., Pontil, M.: Learning the kernel function via regularization. Journal of Machine Learning Research 6, 1099–1125 (2005)
21. Narcowich, F.J., Ward, J.D., Wendland, H.: Sobolev bounds on functions with scattered zeros, with applications to radial basis function surface fitting. Mathematics of Computation 74, 743–763 (2005)
22. Poggio, T., Smale, S.: The Mathematics of Learning: Dealing with Data. Notices of the AMS 50, 537–544 (2003)
23. Raju, I., Phillips, D., Krishnamurthy, T.: A radial basis function approach in the meshless local Petrov-Galerkin method for Euler-Bernoulli beam problems. Computational Mechanics 34, 464–474 (2004)
24. Rieger, C.: Sampling Inequalities and Applications. PhD thesis, Universität Göttingen (2008)
25. Rieger, C., Zwicknagl, B.: Deterministic error analysis of kernel-based regression and related kernel based algorithms. To appear in Journal of Machine Learning Research (2009)
26. Rieger, C., Zwicknagl, B.: Sampling inequalities for infinitely smooth functions, with applications to interpolation and machine learning. To appear in Advances in Computational Mathematics (2009), doi:10.1007/s10444-008-9089-0
27. Schaback, R.: Convergence of Unsymmetric Kernel-Based Meshless Collocation Methods. SIAM J. Numer. Anal. 45, 333–351 (2007)

28. Schaback, R.: Recovery of functions from weak data using unsymmetric meshless kernel-based methods. Applied Numerical Mathematics 58, 726–741 (2007)
29. Schaback, R., Wendland, H.: Inverse and saturation theorems for radial basis function interpolation. Math. Comp. 71, 669–681 (2002)
30. Schaback, R., Wendland, H.: Kernel techniques: from machine learning to meshless methods. Acta Numerica 15, 543–639 (2006)
31. Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press, Cambridge (2002)
32. Schölkopf, B., Williamson, R.C., Bartlett, P.L.: New Support Vector Algorithms. Neural Computation 12, 1207–1245 (2000)
33. Smale, S., Zhou, D.: Estimating the approximation error in learning theory. Analysis and Applications 1, 1–25 (2003)
34. Vapnik, V.: The nature of statistical learning theory. Springer, New York (1995)
35. Wahba, G.: Spline Models for Observational Data. CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 59. SIAM, Philadelphia (1990)
36. Wendland, H.: On the convergence of a general class of finite volume methods. SIAM Journal of Numerical Analysis 43, 987–1002 (2005)
37. Wendland, H.: Scattered Data Approximation. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge (2005)
38. Wendland, H., Rieger, C.: Approximate interpolation with applications to selecting smoothing parameters. Numer. Math. 101, 643–662 (2005)
39. Wu, Z.M.: Hermite–Birkhoff interpolation of scattered data by radial basis functions. Approximation Theory and its Applications 8, 1–10 (1992)
40. Zhang, Y.: Convergence of meshless Petrov-Galerkin method using radial basis functions. Applied Mathematics and Computation 183, 307–321 (2006)
41. Zwicknagl, B.: Power series kernels. Constructive Approximation 29, 61–84 (2009)

Shape Preserving Hermite Interpolation by Rational Biquadratic Splines

Paul Sablonnière

INSA de Rennes, 20 avenue des Buttes de Coësmes, CS 14315,
F-35043 Rennes Cédex, France
psablonn@insa-rennes.fr

Abstract. We construct a rational biquadratic spline interpolating an arbitrary function and its gradient at vertices of a rectangular grid of a domain $\Omega = [a,b] \times [c,d]$. The introduction of the control net allows to give sufficient conditions ensuring the bimonotonicity or the biconvexity of the underlying surface.

Keywords: Bivariate interpolation, bivariate rational splines.

1 Introduction

Let $\mathscr{X}_m := \{a = x_0 < x_2 < \ldots < x_{2m} = b\}$ be a partition of $[a,b]$, and $\mathscr{Y}_n := \{c = y_0 < y_2 < \ldots < y_{2n} = d\}$ a partition of $[c,d]$. The rectangular domain $\Omega = [a,b] \times [c,d]$ is endowed with the induced tensor-product partition $\mathscr{Z}_{mn} := \mathscr{X}_m \otimes \mathscr{Y}_n$. Given Hermite data at the vertices $A_{i,j} := (x_{2i}, y_{2j})$:

$$(f_{i,j}, p_{i,j}, q_{i,j}), \quad \text{with} \quad f_{i,j} := f(A_{i,j}), \;\; p_{i,j} := \frac{\partial f}{\partial x}(A_{i,j}) \;\; q_{i,j} := \frac{\partial f}{\partial y}(A_{i,j}),$$

we construct a rational biquadratic spline S interpolating those data. More specifically, we consider the subpartition $\mathscr{Z}'_{mn}$ of $\mathscr{Z}_{mn}$ obtained by subdividing each rectangular cell $\Omega_{ij} := [x_{2i-2}, x_{2i}] \times [y_{2j-2}, y_{2j}]$ into four equal subrectangles. In each cell of $\mathscr{Z}'_{mn}$, S is a rational function with *biquadratic numerator* and *bilinear denominator*. In addition, we prove that S preserves the *bimonotonicity* (resp. the *biconvexity*) of the data, i.e. the monotonicity (resp. convexity) in the directions of both coordinate axes.

In Section 2, we recall some results on two-point Hermite interpolation by shape preserving univariate rational quadratic splines. In Sections 3 and 4, we extend these results to shape preserving properties of tensor product interpolants on rectangular cells Ω_{ij}. In Section 5, we give simple examples illustrating our method. Then, in Section 6, we extend the method to Hermite interpolation over the whole domain Ω. Finally, in Section 7, we define tensor-product B-splines allowing the global representation of the piecewise rational interpolant.

Our results are quite similar to those obtained for example in [13] and [21] on tensor products of Hermite subdivision schemes (see also [11] and [18]) and on variable degree [3] [13] [2] [4] or parametric [6] [19] splines. It would be interesting to test blending schemes, as done in [5] [7] for bivariate splines with variable degrees. Like most of the previous papers, they extend previous works by Carlson and Fritsch [1] [14],

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 370–384, 2010.

mainly based on bicubics, since *arbitrary values* of partial derivatives are allowed. The idea of using rational quadratic splines in the univariate case dates back to Delbourgo and Gregory [8] [17] [9] [15] [16]. In addition, at the author's knowledge, there is no paper on bivariate shape-preserving rational splines. However, the introduction of *control polygons and control nets* in the study of rational interpolants seems to be new. I Anyway, it greatly simplifies the study of monotonicity and convexity properties of these types of curves and surfaces.

2 Univariate C^1 Rational Quadratic Splines

We define a **rational quadratic spline** R in the interval $I = [-1,1]$ as a piecewise rational quadratic function whose restrictions to the subintervals $I_0 := [-1,0]$ and $I_1 := [0,1]$ are respectively

$$R(u) := \frac{\widetilde{P}(u)}{\widetilde{Q}(u)} = \frac{w_0 c_0 (1-u)^2 + (w_0+w)c_1 u(1-u) + w c_2 u^2}{w_0(1-u)+wu}, \quad x \in I_0,\ u = x+1,$$

$$R(x) := \frac{\overline{P}(x)}{\overline{Q}(x)} = \frac{w d_0 (1-x)^2 + (w+w_1)d_1 x(1-x) + w_1 d_2 x^2}{(1-x)w + w_1 x}, \quad x \in I_1.$$

The main weights are w_0 and w_1, the third weight w being adjusted in function of the latter. The C^1 continuity of R at the origin is expressed by

$$c_2 = d_0 = \alpha_0 c_1 + \alpha_1 d_1,$$

where $\alpha_0 := \dfrac{w_0+w}{w_0+2w+w_1}$ and $\alpha_1 := 1-\alpha_0 = \dfrac{w_1+w}{w_0+2w+w_1}$.

2.1 First and Second Derivatives

The **first derivative** of R is given, respectively in each subinterval, by

$$R'(u) = \frac{\widetilde{P}_1(u)}{\widetilde{Q}^2(u)} \quad \text{and} \quad R'(x) = \frac{\overline{P}_1(x)}{\overline{Q}^2(x)}.$$

Using the positive functions σ_0 and σ_1 respectively defined in the quadratic Bernstein basis $\{B_0, B_1, B_2\}$ of each subinterval by

$$\widetilde{\sigma}_0(u) = w_0\left((w+w_0)B_0(u) + wB_1(u)\right), \quad \widetilde{\sigma}_1(u) = w\left((w+w_0)B_2(u) + w_0 B_1(u)\right),$$

$$\overline{\sigma}_0(x) = w\left((w+w_1)B_0(x) + w_1 B_1(x)\right), \quad \overline{\sigma}_1(x) = w_1\left((w+w_1)B_2(x) + wB_1(x)\right),$$

the numerators can be written in the convenient form

$$\widetilde{P}_1(u) = \Delta c_0\, \widetilde{\sigma}_0(u) + \Delta c_1\, \widetilde{\sigma}_1(u), \qquad \overline{P}_1(x) = \Delta d_0\, \overline{\sigma}_0(x) + \Delta d_1\, \overline{\sigma}_1(x).$$

For the **second derivative** of R, setting

$$\delta_0 := w c_0 - (w+w_0)c_1 + w_0 c_2, \qquad \delta_1 := w_1 d_0 - (w+w_1)d_1 + w d_2,$$

we obtain in each subinterval

$$\widetilde{R}''(u) = \frac{2ww_0\delta_0}{\widetilde{Q}^3(u)}, \qquad \overline{R}''(x) = \frac{2ww_1\delta_1}{\overline{Q}^3(x)}.$$

It is easy to see that these quantities are positively proportional to the *second order divided differences* of the control polygons $\mathbf{c}$ of R_0 and $\mathbf{d}$ of R_1:

$$[-1,-1+\theta_0,0]\,\mathbf{c} = \frac{w+w_0}{w_0}\,\delta_0, \qquad [0,1-\theta_1,1]\,\mathbf{d} = \frac{w+w_1}{w_1}\,\delta_1.$$

2.2 Hermite Interpolant

It is now easy to construct a rational quadratic spline R solution of the local Hermite interpolation problem:

$$R(-1) = f_0, \quad R'(-1) = p_0, \quad R(1) = f_1, \quad R'(1) = p_1.$$

From the preceding section, we deduce

$$c_0 = f_0, \qquad d_2 = f_1, \qquad \frac{c_1 - c_0}{\theta_0} = p_0, \qquad \frac{d_2 - d_1}{\theta_1} = p_1,$$

therefore we get

$$c_1 = f_0 + \theta_0 p_0 = f_0 + \frac{w_0\,p_0}{w_0+w}, \quad d_1 = f_1 - \theta_1 p_1 = f_1 - \frac{w_1 p_1}{w_1+w},$$

and we see that there exists a *unique solution* depending on the choice of the three positive weights w_0, w_1, w.

2.3 Control Polygon

According to the results in [20] [22], setting

$$\theta_0 = \frac{w_0}{w_0+w} \quad \text{and} \quad \theta_1 = \frac{w_1}{w_1+w},$$

then the **control polygon** of R on I_0 has vertices:

$$\mathbf{c}_0 = (-1, c_0), \quad \mathbf{c}_1 = (-1+\theta_0, c_1) \quad \mathbf{c}_2 = (0, c_2),$$

and that of R on I_1 has vertices:

$$\mathbf{d}_0 = (0, d_0), \quad \mathbf{d}_1 = (1-\theta_1, d_1), \quad \mathbf{d}_2 = (1, d_2).$$

As $\mathbf{c}_2 = \mathbf{d}_0 = \alpha_0\mathbf{c}_1 + \alpha_1\mathbf{d}_1$, we see that the rational spline R is completely determined by its four control points $\{\mathbf{c}_0, \mathbf{c}_1, \mathbf{d}_1, \mathbf{d}_2\}$. The results of Section 2.1 allow to prove the following

Theorem 1. *R is monotone (resp. convex) if and only if its control polygon is monotone (resp. convex).*

2.4 Rational Bernstein and Hermite Bases

The Bernstein basis $\mathscr{B} := \{\rho_0, \rho_1, \rho_2, \rho_3\}$ is composed of the four rational quadratic splines defined respectively by $\widetilde{\rho}_i(u) = \widetilde{\pi}_i(u)/\widetilde{Q}(u)$ for $x = u - 1 \in I_0$ and $\overline{\rho}_i(x) = \overline{\pi}_i(x)/\overline{Q}(x)$ for $x \in I_1$, where:

$$\begin{array}{cc} \widetilde{\pi}_0(u) = w_0(1-u)^2 & \overline{\pi}_0(x) = 0 \\ \widetilde{\pi}_1(u) = (w_0+w)u(1-u) + \alpha_0 w u^2 & \overline{\pi}_1(x) = \alpha_0 w(1-x)^2 \\ \widetilde{\pi}_2(u) = \alpha_1 w u^2 & \overline{\pi}_2(x) = \alpha_1 w(1-x)^2 + (w+w_1)x(1-x) \\ \widetilde{\pi}_3(u) = 0 & \overline{\pi}_3(x) = w_1 x^2 \end{array}$$

It is easy to verify that these functions satisfy the interpolation properties:

$$\rho_0(-1) = \rho_3(1) = 1; \quad \rho_1(-1) = \rho_2(-1) = \rho_3(-1) = \rho_0(1) = \rho_1(1) = \rho_2(1) = 0.$$

$$-\rho_0'(-1) = \rho_1'(-1) = \frac{1}{\theta_0}, \quad -\rho_2'(1) = \rho_3'(1) = \frac{1}{\theta_1},$$
$$\rho_2'(-1) = \rho_3'(-1) = \rho_0'(1) = \rho_1'(1) = 0.$$

Moreover they form a *positive partition of unity* and a *totally positive system* [22]. It is now easy to derive from $\mathscr{B}$ the *Hermite basis* $\mathscr{H} := \{\varphi_0, \psi_0, \varphi_1, \psi_1\}$ associated with the interpolation problem. The four rational splines in $\mathscr{H}$ have control polygons whose ordinates are respectively

$$(1,1,0,0), \quad (0,\theta_0,0,0), \quad (0,0,1,1), \quad (0,0,-\theta_1,0),$$

the abscissae being $(-1, -1+\theta_0, 1-\theta_1, 1)$. In other words, we have:

$$\varphi_0 = \rho_0 + \rho_1, \quad \psi_0 = \theta_0\rho_1, \quad \varphi_1 = \rho_2 + \rho_3, \quad \psi_1 = -\theta_1\rho_2.$$

Then, the solution of the Hermite interpolation problem can be expressed in that basis

$$R = y_0\varphi_0 + p_0\psi_0 + y_1\varphi_1 + p_1\psi_1.$$

2.5 Monotone Rational Quadratic Splines

Consider the Hermite problem with increasing data: $f_0 < f_1$, $p_0 > 0$, $p_1 > 0$. The monotonicity of the rational quadratic spline is *equivalent* (Theorem 1) to that of its control polygon, i.e. to the inequality:

$$(M) \qquad p_0\theta_0 + p_1\theta_1 < \Delta f_0 := f_1 - f_0.$$

The problem admits an infinite number of solutions. Here is a simple algorithm providing such a solution.

Algorithm (M)
1. if $p_0 + p_1 \le 2\Delta f_0$, then choose $w_0 = w_1 = w = 1$ (quadratic spline interpolant).
2. if $p_0 + p_1 > 2\Delta f_0$, then choose $w_0 = \Delta f_0/(2p_0), w_1 = \Delta f_0/(2p_1)$ and $w = 1$.

Then inequality (M) is automatically satisfied and the interpolant is monotone.

2.6 Convex Rational Quadratic Splines

Consider the case of convex data $(f_0,p_0;f_1,p_1) : p_0 < \Delta f_0/2 < p_1$. By Theorem 1, the function R is convex *if and only if* its control polygon is convex, i.e.

$$(C) \qquad p_0 \le \frac{\Delta f_0 - p_0\theta_0 - p_1\theta_1}{2-\theta_0-\theta_1} \le p_1,$$

which is equivalent to both inequalities:

$$\frac{w_0}{w_0+w} = \theta_0 \le \frac{2p_1-\Delta f_0}{p_1-p_0}, \quad \frac{w_1}{w_1+w} = \theta_1 \le \frac{\Delta f_0 - 2p_0}{p_1-p_0}.$$

The problem admits an infinity of solutions. Here is a simple algorithm providing such a solution.

Algorithm (C)
Choose $w_0 = 2p_1 - \Delta f_0$, $w_1 = \Delta f_0 - 2p_0$, $w \ge |p_0+p_1-\Delta f_0|$.
Then inequality (C) *is automatically satisfied and the interpolant is convex.*

3 Tensor Product of Rational Splines

In the square $\Omega := I \times J = [-1,1]^2$, a tensor-product (abbr. TP) rational biquadratic spline S can be expressed either in the Hermite basis or in the Bernstein basis. We denote by $\{\varphi_0,\psi_0,\varphi_1,\psi_1\}$ (resp. $\{\rho_0,\rho_1,\rho_2,\rho_3\}$) the Hermite (resp. Bernstein) basis w.r.t. the variables $(u,x) \in I_0 \times I_1$ and by $\{\widetilde{\varphi}_0,\widetilde{\psi}_0,\widetilde{\varphi}_1,\widetilde{\psi}_1\}$ (resp. $\{\widetilde{\rho}_0,\widetilde{\rho}_1,\widetilde{\rho}_2,\widetilde{\rho}_3\}$) the Hermite (resp. Bernstein) basis w.r.t. the variables $(v,y) \in I_0 \times I_1$. We set $f_{0,0} := f(-1,-1), f_{0,1} := f(-1,1), f_{1,0} := f(1,-1), f_{1,1} := f(1,1)$ and for the derivatives

$$p_{i,j} := D^{1,0}f_{i,j} = \frac{\partial}{\partial x}f_{i,j}, \quad q_{i,j} := D^{0,1}f_{i,j} = \frac{\partial}{\partial y}f_{i,j}, \quad r_{i,j} := D^{1,1}f_{i,j} = \frac{\partial^2}{\partial x \partial y}f_{i,j}.$$

The weights on I (resp. J) being denoted $\{w_0,w_1,w\}$ (resp. $\{\omega_0,\omega_1,\omega\}$), we set

$$\theta_0 := \frac{w_0}{w_0+w}, \quad \theta_1 := \frac{w_1}{w_1+w}, \qquad \tau_0 := \frac{\omega_0}{\omega_0+\omega}, \quad \tau_1 := \frac{\omega_1}{\omega_1+\omega}.$$

$$\beta_0 := \frac{\omega_0+\omega}{\omega_0+2\omega+\omega_1}, \quad \beta_1 := 1-\beta_0 = \frac{\omega+\omega_1}{\omega_0+2\omega+\omega_1},$$

the parameters α_0 and α_1 being defined in Section 2. Of course, all functions in the variable y (or $v = y+1$) must be computed with the corresponding weights $(\omega_0,\omega_1,\omega)$.

Since the **TP Hermite basis** is defined by $\varphi_{i,j} := \varphi_i\widetilde{\varphi}_j$, $\xi_{i,j} := \psi_i\widetilde{\varphi}_j$, $\eta_{i,j} := \varphi_i\widetilde{\psi}_j$, $psi_{i,j} := \psi_i\widetilde{\psi}_j$, the expression of S is

$$S = \sum_{i=0}^{1}\sum_{j=0}^{1}\left(f_{i,j}\varphi_{i,j} + p_{i,j}\xi_{i,j} + q_{i,j}\eta_{i,j} + r_{i,j}\psi_{i,j}\right).$$

As we have respectively $\varphi_0 = \rho_0 + \rho_1$, $\psi_0 = \theta_0\rho_1$, $\varphi_1 = \rho_2 + \rho_3$, $\psi_1 = -\theta_1\rho_1$, and $\widetilde{\varphi}_0 = \widetilde{\rho}_0 + \widetilde{\rho}_1$, $\widetilde{\psi}_0 = \tau_0\widetilde{\rho}_1$, $\widetilde{\varphi}_1 = \widetilde{\rho}_2 + \widetilde{\rho}_3$, $\widetilde{\psi}_1 = -\tau_1\widetilde{\rho}_2$, we obtain for example

$$\begin{aligned} \varphi_{0,0} &= (\rho_0 + \rho_1)(\widetilde{\rho}_0 + \widetilde{\rho}_1), & \eta_{0,0} &= (\rho_0 + \rho_1)\tau_0\widetilde{\rho}_1, \\ \xi_{0,0} &= \theta_0\rho_1(\widetilde{\rho}_0 + \widetilde{\rho}_1), & \psi_{0,0} &= \theta_0\rho_1\tau_0\widetilde{\rho}_1. \end{aligned}$$

The expression $S_{0,0} := f_{0,0}\,\varphi_{0,0} + p_{0,0}\xi_{0,0} + q_{0,0}\eta_{0,0} + r_{0,0}\psi_{0,0}$ therefore becomes

$$\begin{aligned} S_{0,0} = f_{0,0}\,\rho_0\widetilde{\rho}_0 + (f_{0,0} + \tau_0 q_{0,0})\,\rho_0\widetilde{\rho}_1 + \\ (f_{0,0} + \theta_0 p_{0,0})\,\rho_1\widetilde{\rho}_0 + (f_{0,0} + \theta_0 p_{0,0} + \tau_0 q_{0,0} + \theta_0\tau_0 r_{0,0})\,\rho_1\widetilde{\rho}_1. \end{aligned}$$

Similar expressions hold for the three other restrictions $S_{1,0}, S_{0,1}, S_{1,1}$ of S.

The expression of S in the **TP Bernstein basis** is

$$S = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{i,j}\rho_{i,j}, \quad \text{with} \quad \rho_{i,j}(x,y) = \rho_i(x)\widetilde{\rho}_j(y).$$

The **control net** of S is formed by the 16 points of the space whose coordinates are respectively $(x_i, y_j, a_{i,j})$, the Greville coordinates being given by

$$(x_0,x_1,x_2,x_3) := (-1, -1+\theta_0, 1-\theta_1, 1), \quad (y_0,y_1,y_2,y_3) := (-1, -1+\tau_0, 1-\tau_1, 1).$$

Comparing $S_{0,0} = \sum_{i=0}^{1}\sum_{j=0}^{1} a_{i,j}\rho_{i,j}$ with its expression given above, we get

$$\begin{aligned} &a_{0,0} = f_{0,0}, \quad a_{1,0} = f_{0,0} + \theta_0 p_{0,0}, \quad a_{0,1} = f_{0,0} + \tau_0 q_{0,0}, \\ &a_{1,1} = f_{0,0} + \theta_0 p_{0,0} + \tau_0 q_{0,0} + \theta_0\tau_0 r_{0,0}, \end{aligned}$$

and similar expressions for the BB-coefficients at the other vertices,

$$\begin{aligned} &a_{3,0} = f_{1,0}, \quad a_{2,0} = f_{1,0} - \theta_1 p_{1,0}, \quad a_{3,1} = f_{1,0} + \tau_0 q_{1,0}, \\ &a_{2,1} = f_{1,0} - \theta_1 p_{1,0} + \tau_0 q_{1,0} - \theta_1\tau_0 r_{1,0}, \\ &a_{0,3} = f_{0,1}, \quad a_{1,3} = f_{0,1} + \theta_0 p_{0,1}, \quad a_{0,2} = f_{0,1} - \tau_1 q_{0,1}, \\ &a_{1,2} = f_{0,1} + \theta_0 p_{0,1} - \tau_1 q_{0,1} - \theta_0\tau_1 r_{0,1}, \\ &a_{3,3} = f_{1,1}, \quad a_{2,3} = f_{1,1} - \theta_1 p_{1,1}, \quad a_{3,2} = f_{1,1} - \tau_1 q_{1,1}, \\ &a_{2,2} = f_{1,1} - \theta_1 p_{1,1} - \tau_1 q_{1,1} + \theta_1\tau_1 r_{1,1}. \end{aligned}$$

4 Shape Preserving Properties

4.1 Sufficient Conditions on the Control Net

Theorem 2. *If its control net is bimonotone in x and y, then the surface S is also bimonotone in x and y.*

Proof. Setting $\Delta_1 a_{i,j} = a_{i+1,j} - a_{i,j}$, $\Delta_2 a_{i,j} = a_{i,j+1} - a_{i,j}$ and using the results of Section 2.1, we obtain the following expressions:

$$\frac{\partial S}{\partial x} = \sum_{j=0}^{3} \left(\Delta_1 a_{0,j}\, \widetilde{\sigma}_0(u) + \alpha_1 \Delta_1 a_{1,j}\, \widetilde{\sigma}_1(u)\right) \tilde{\rho}_j(y)/\widetilde{Q}^2(u), \quad x \in I_0, u = x+1,$$

$$\frac{\partial S}{\partial x} = \sum_{j=0}^{3} \left(\alpha_0 \Delta_1 a_{1,j}\, \overline{\sigma}_0(x) + \Delta_1 a_{2,j}\, \overline{\sigma}_1(x)\right) \tilde{\rho}_j(y)/\overline{Q}^2(x), \quad x \in I_1,$$

$$\frac{\partial S}{\partial y} = \sum_{i=0}^{3} \left(\Delta_2 a_{i,0}\, \widetilde{\sigma}_0(v) + \beta_1 \Delta_2 a_{i,1}\, \widetilde{\sigma}_0(v)\right) \rho_i(x)/\widetilde{Q}^2(v) \quad y \in J_0 = [-1,0], v = y+1,$$

$$\frac{\partial S}{\partial y} = \sum_{i=0}^{3} \left(\beta_0 \Delta_2 a_{i,1}\, \overline{\sigma}_0(y) + \Delta_2 a_{i,2}\, \overline{\sigma}_1(y)\right) \rho_i(x)\overline{Q}^2(y), \quad y \in J_1 = [0,1],$$

where the functions σ_i and σ_j (and their restrictions to both subintervals) are positive. Therefore, it is clear that the positivity of the first differences of coefficients implies that of partial derivatives, hence the partial monotonicities in x and y.

Theorem 3. *If its control net is biconvex in x and y, then the surface S is also biconvex in x and y.*

Proof. As in Theorem 2, we use the expressions of second partial derivatives derived from those of Section 2.1, respectively for $x \in I_0\,(u = x+1)$ and $x \in I_1$:

$$\frac{\partial^2 S}{\partial x^2} = \kappa_0 \sum_{j=0}^{3} \frac{\delta_{0,j}}{\widetilde{Q}^3(u)} \tilde{\rho}_j(y), \quad \text{resp.} \quad \kappa_1 \sum_{j=0}^{3} \frac{\delta_{1,j}}{\overline{Q}^3(x)} \tilde{\rho}_j(y),$$

and for $y \in J_0\,(v = y+1)$ and $y \in J_1$:

$$\frac{\partial^2 S}{\partial y^2} = \sum_{i=0}^{3} \frac{\delta_{i,0}}{\widetilde{Q}^3(v)} \rho_i(x), \quad \text{resp.} \quad \kappa_4 \sum_{i=0}^{3} \frac{\delta_{i,1}}{\overline{Q}^3(y)} \rho_i(x),$$

where the $\kappa'_\ell s$ are positive coefficients depending only on the weights and

$$\delta_{0,j} := \alpha_1 w_0 \Delta_1 a_{1,j} - w \Delta_1 a_{0,j} \qquad \delta_{1,j} := -\alpha_0 w_1 \Delta_1 a_{1,j} + w \Delta_1 a_{2,j},$$

$$\delta_{i,0} := \beta_1 \omega_0 \Delta_2 a_{i,1} - \omega \Delta_2 a_{i,0} \qquad \delta_{i,1} := -\beta_0 \omega_1 \Delta_2 a_{i,1} + \omega \Delta_2 a_{i,2}.$$

These quantities are positively proportional to second order divided differences (abbr. dd) of the control net in both directions. For example, $\delta_{0,j}$ is proportional to the dd of the control points $(-1, a_{0,j}), (-1+\theta_0, a_{1,j}), (1-\theta_1, a_{2,j})$, and similar expressions for the other δ's. Thus, the positivity of the dd in both directions x and y (i.e. the biconvexity of the control net) implies that of second order partial derivatives.

4.2 Bimonotonicity of the Local Control Net

Computing the differences $\Delta_1 a_{i,j}$ (resp. $\Delta_2 a_{i,j}$), we can see that the *monotonicity* in x (resp. in y) of the control net needs 16 inequalities to be satisfied, other than the inequalities $p_{i,j}, q_{i,j} \geq 0$. For the sake of clearness, *the data are assumed to be increasing in both directions*. We use the following notations

$$\delta_1 := \Delta_1 f_{00} \geq 0,\ \overline{\delta}_1 := \Delta_1 f_{01} \geq 0,\ \delta_2 := \Delta_2 f_{00} \geq 0,\ \overline{\delta}_2 := \Delta_2 f_{10} \geq 0,$$

$$\zeta_1 := \Delta_1 p_{00}, \quad \overline{\zeta}_1 := \Delta_1 p_{01}, \quad \zeta_2 := \Delta_2 p_{00}, \quad \overline{\zeta}_2 := \Delta_2 p_{10},$$

$$\eta_1 := \Delta_1 q_{00}, \quad \overline{\eta}_1 := \Delta_1 q_{01}, \quad \eta_2 := \Delta_2 q_{00}, \quad \overline{\eta}_2 := \Delta_2 q_{10}.$$

Then, the inequalities are the following:

$$\begin{aligned}
p_{00}\theta_0 + p_{10}\theta_1 &\leq \delta_1, & p_{00}\theta_0 + p_{10}\theta_1 &\leq \delta_1 + (\eta_1 - (r_{00}\theta_0 + r_{10}\theta_1))\tau_0,\\
p_{01}\theta_0 + p_{11}\theta_1 &\leq \overline{\delta}_1, & p_{01}\theta_0 + p_{11}\theta_1 &\leq \overline{\delta}_1 + (r_{01}\theta_0 + r_{11}\theta_1 - \overline{\eta}_1)\tau_1,\\
q_{00}\tau_0 + q_{01}\tau_1 &\leq \delta_2, & q_{00}\tau_0 + q_{01}\tau_1 &\leq \delta_2 + (\zeta_2 - (r_{00}\tau_0 + r_{01}\tau_1))\theta_0,\\
q_{10}\tau_0 + q_{11}\tau_1 &\leq \overline{\delta}_2, & q_{10}\tau_0 + q_{11}\tau_1 &\leq \overline{\delta}_2 + (r_{10}\tau_0 + r_{11}\tau_1 - \overline{\zeta}_2)\theta_1.
\end{aligned}$$

$$\begin{aligned}
&p_{00} + r_{00}\tau_0 \geq 0, \quad && p_{10} + r_{10}\tau_0 \geq 0, \quad && p_{01} - r_{01}\tau_1 \geq 0, \quad && p_{11} - r_{11}\tau_1 \geq 0,\\
&q_{00} + r_{00}\theta_0 \geq 0, \quad && q_{01} + r_{01}\theta_0 \geq 0, \quad && q_{10} - r_{10}\theta_1 \geq 0, \quad && q_{11} - r_{11}\theta_1 \geq 0,
\end{aligned}$$

In view of the positivity of the slopes δ_i, $\overline{\delta}_j$ and of partial derivatives, it is always possible to find sufficiently small values of the four parameters $\theta_0, \theta_1, \tau_0, \tau_1$ so that the above inequalities are satisfied. Considering these parameters as functions of the weights $\{w_0, w, w_1, \omega_0, \omega, \omega_1\}$ we get:

Theorem 4. *It is always possible to find values of the weights so that the control net of the rational spline interpolant is increasing both in x and y. Therefore the interpolant itself has the same property.*

4.3 Biconvexity of the Local Control Net

We assume that the data are convex in both directions, thus they satisfy the following inequalities.

$$\zeta_1 > 0, \quad \overline{\zeta}_1 > 0, \quad \eta_2 > 0, \quad \overline{\eta}_2 > 0.$$

Using the additional notations

$$\rho_1 := \Delta_1 r_{00}, \quad \overline{\rho}_1 := \Delta_1 r_{01}, \quad \rho_2 := \Delta_2 r_{00}, \quad \overline{\rho}_2 := \Delta_2 r_{10},$$

and computing the slopes of the control net in both directions, we get the 16 following inequalities for the biconvexity of the control net:

$$\begin{aligned}
\zeta_1\theta_0 &\le 2p_{10}-\delta_1 & \zeta_1\theta_0 &\le (2p_{10}-\delta_1)-(\eta_1-2r_{10}+\rho_1\theta_0)\tau_0\\
\overline{\zeta}_1\theta_0 &\le 2p_{11}-\overline{\delta}_1 & \overline{\zeta}_1\theta_0 &\le (2p_{11}-\overline{\delta}_1)+(\overline{\eta}_1-2r_{11}+\overline{\rho}_1\theta_0)\tau_1\\
\zeta_1\theta_1 &\le \delta_1-2p_{00} & \zeta_1\theta_1 &\le (\delta_1-2p_{00})+(\eta_1-2r_{00}-\rho_1\theta_1)\tau_0\\
\overline{\zeta}_1\theta_1 &\le \overline{\delta}_1-2p_{01} & \overline{\zeta}_1\theta_1 &\le (\overline{\delta}_1-2p_{01})-(\overline{\eta}_1-2r_{01}-\overline{\rho}_1\theta_1)\tau_1\\
\eta_2\tau_0 &\le 2q_{01}-\delta_2 & \eta_2\tau_0 &\le (2q_{01}-\delta_2)-(\zeta_2-2r_{01}+\rho_2\tau_0)\theta_0\\
\overline{\eta}_2\tau_0 &\le 2q_{11}-\overline{\delta}_2 & \overline{\eta}_2\tau_0 &\le (2q_{11}-\overline{\delta}_2)+(\overline{\zeta}_2-2r_{11}+\overline{\rho}_2\tau_0)\theta_1\\
\eta_2\tau_1 &\le \delta_2-2q_{00} & \eta_2\tau_1 &\le (\delta_2-2q_{00})+(\zeta_2-2r_{00}-\rho_2\tau_1)\theta_0\\
\overline{\eta}_2\tau_1 &\le \overline{\delta}_2-2q_{10} & \overline{\eta}_2\tau_1 &\le (\overline{\delta}_2-2q_{10})-(\overline{\zeta}_2-2r_{10}-\overline{\rho}_2\tau_1)\theta_1
\end{aligned}$$

As the right-hand sides of left inequalities and the quantities $\zeta_1, \overline{\zeta}_1, \eta_2, \overline{\eta}_2$ are all positive, it is always possible to find sufficiently small values of the four parameters θ_0, θ_1, τ_0, τ_1 so that the above inequalities are satisfied. Considering these parameters as functions of the weights $\{w_0, w, w_1, \omega_0, \omega, \omega_1\}$, we get:

Theorem 5. *It is always possible to find values of the weights so that the control net of the rational spline interpolant is convex, both in x and y. Therefore the interpolant itself has the same property.*

4.4 Algorithms

As pointed out by one of the referees, it would be nice to have simple algorithms like Algorithm (M) (Section 2.5) and Algorithm (C) (Section 2.6) for surfaces. However, the problem has now become much more complex, since it depends on the 6 weights $\{w_0, w, w_1, \omega_0, \omega, \omega_1\}$ plus the 4 crossed derivatives $\{r_{00}, r_{10}, r_{01}, r_{11}\}$ that can also be considered as additional shape parameters.

5 Examples

We consider two sets of Hermite data, the first for bimonotonicity, the second for biconvexity. (The first is similar to a specific one extracted from the classical example of the aluminium data set [1] [3]).

	f_{00}	f_{10}	f_{01}	f_{11}	p_{00}	p_{10}	p_{01}	p_{11}	q_{00}	q_{10}	q_{01}	q_{11}	r_{00}	r_{10}	r_{01}	r_{11}
$Ex1$	0	64	62	65	0	20	3	18	90	16	3	2	0	-16	3	2
$Ex2$	0	0	0	0	-20	40	-3	48	-2	-13	50	1	10	-3	-2	8

With the notations of Section 4.3, we compute the following table:

	δ_1	$\overline{\delta}_1$	δ_2	$\overline{\delta}_2$	ζ_1	$\overline{\zeta}_1$	ζ_2	$\overline{\zeta}_2$	η_1	$\overline{\eta}_1$	η_2	$\overline{\eta}_2$	ρ_1	$\overline{\rho}_1$	ρ_2	$\overline{\rho}_2$
$Ex1$	64	3	62	1	20	15	3	-2	-74	-1	-87	-14	-16	-1	3	18
$Ex2$	0	0	0	0	60	51	17	8	-11	-49	52	14	-13	10	-12	11

Example 6. We notice that the data are increasing in both directions. For the *bimonotonicity* of the surface, the coefficients $\theta_0, \theta_1, \tau_0, \tau_1$ must satisfy the following inequalities

$$\begin{aligned} 5\theta_1 &\le 16 & 5\theta_1 &\le 16+(8\theta_1-37)\tau_0/2 \\ \theta_0+6\theta_1 &\le 1 & \theta_0+6\theta_1 &\le 1+(3\theta_0+2\theta_1+1)\tau_1/3 \\ 30\tau_0+\tau_1 &\le 62/3 & 30\tau_0+\tau_1 &\le 62/3+(1-\tau_1)\theta_0 \\ 8\tau_0+\tau_1 &\le 1/2 & 8\tau_0+\tau_1 &\le 1/2+(-8\tau_0+\tau_1+1)\theta_1 \end{aligned}$$

The solution $\theta_0=\theta_1=\tau_0=\tau_1=\frac{1}{2}$ is not admissible, so classical biquadratic splines do not work. An admissible solution is $(\theta_0,\theta_1,\tau_0,\tau_1)=\left(\frac{1}{4},\frac{1}{8},\frac{1}{32},\frac{1}{4}\right)$: it gives some equalities and all inequalities are satisfied. An associated set of admissible weights is for example $(w_0,w,w_1;\omega_0,\omega,\omega_1)=(7,21,3;3,93,31)$.

Example 7. For the *biconvexity* of the surface, the coefficients $\theta_0, \theta_1, \tau_0, \tau_1$ must satisfy the following inequalities

$$\begin{aligned} 60\theta_0 &\le 80 & 60\theta_0 &\le 80+(5+13\theta_0)\tau_0 \\ 51\theta_0 &\le 96 & 51\theta_0 &\le 96-(65-10\theta_0)\tau_1 \\ 60\theta_1 &\le 40 & 60\theta_1 &\le 40-(31-13\theta_1)\tau_0 \\ 51\theta_1 &\le 6 & 51\theta_1 &\le 6+(45+10\theta_1)\tau_1 \\ 52\tau_0 &\le 100 & 52\tau_0 &\le 100-(21-12\tau_0)\theta_0 \\ 14\tau_0 &\le 2 & 14\tau_0 &\le 2-(8-11\tau_0)\theta_1 \\ 52\tau_1 &\le 4 & 52\tau_1 &\le 4-(3-12\tau_1)\theta_0 \\ 14\tau_1 &\le 26 & 14\tau_1 &\le 26-(14-11\tau_1)\theta_1 \end{aligned}$$

In particular, we get the constraints

$$\theta_1 \le 2/17, \quad \tau_0 \le 1/7, \quad \tau_1 \le 1/13.$$

	θ_0	θ_1	τ_0	τ_1	w_0	w	w_1	ω_0	ω	ω_1
*Ex*1	1/4	1/8	1/32	1/4	7	21	3	3	93	31
*Ex*2	1/3	1/17	1/15	1/16	15	30	4	15	210	14

The solution $\theta_0=\theta_1=\tau_0=\tau_1=\frac{1}{2}$ is not admissible, so classical biquadratic splines do not work. If we try

$$\theta_1 = 2/17, \quad \tau_0 = 1/7, \quad \tau_1 = 1/13,$$

then the two following inequalities are not satisfied:

$$(14-11\theta_1)\tau_0 \le 2-8\theta_1 \quad (52-12\theta_0)\tau_1 \le 4-3\theta_0.$$

An admissible solution is $(\theta_0,\theta_1,\tau_0,\tau_1)=\left(\frac{1}{3},\frac{2}{17},\frac{1}{15},\frac{1}{16}\right)$. An associated set of admissible weights is for example $(w_0,w,w_1;\omega_0,\omega,\omega_1)=(15,30,4;15,210,14)$. In both cases, we notice the higher values of the central weights w and ω.

6 Hermite Interpolation in the Whole Domain

Consider a subrectangle $R_{ij} = [x_{2i-2}, x_{2i}] \times [y_{2j-2}, y_{2j}]$ of the partition. The weights (respectively abscissae) corresponding to those denoted $(w_0, w, w_1, \omega_0, \omega, \omega_1)$ (respectively $(\theta_0, \theta_1, \tau_0, \tau_1)$) in the preceding sections are now denoted $(\widetilde{w}_i, w_i, \overline{w}_i, \widetilde{\omega}_j, \omega_j, \overline{\omega}_j)$ (resp. $(\widetilde{\theta}_i, \overline{\theta}_i, \widetilde{\tau}_j, \overline{\tau}_j)$). Setting $h_i := x_{2i+2} - x_{2i}$, $k_j := y_{2j+2} - y_{2j}$, the slopes on the horizontal and vertical edges of R_{ij} are denoted respectively

$$\delta_{ij}^H := (f_{i+1,j} - f_{i,j})/h_i, \qquad \delta_{ij}^V := (f_{i,j+1} - f_{i,j})/k_j.$$

In this section, we assume, for the sake of simplicity and clarity, that *all crossed derivatives r_{ij} are set to zero*. Using the notations:

$$\zeta_{ij}^H := p_{i+1,j} - p_{ij}, \qquad \eta_{ij}^H := q_{i+1,j} - q_{ij},$$
$$\zeta_{ij}^V := p_{i,j+1} - p_{ij}, \qquad \eta_{ij}^V := q_{i,j+1} - q_{ij},$$

we write all the constraints acting on the weights in the subdomain R_{ij}. We only sketch the proofs. More details are needed for writing practical algorithms, they will be developed elsewhere.

6.1 Global Bimonotonicity

Assuming the data are increasing in both directions, we obtain the following inequalities on the parameters $\widetilde{\theta}_i$ and $\overline{\theta}_i$, for all indices j (resp. $\widetilde{\tau}_j$ and $\overline{\tau}_j$, for all indices i) in the vertical (resp. horizontal) strip between $x = x_{2i}$ and $x = x_{2i+2}$ (resp. $y = y_{2j}$ and $y = y_{2j+2}$):

$$p_{ij}\widetilde{\theta}_i + p_{i+1,j}\,\overline{\theta}_i \le 2\delta_{ij}^H \qquad p_{ij}\widetilde{\theta}_i + p_{i+1,j}\,\overline{\theta}_i \le 2\delta_{ij}^H + \frac{k_j}{h_i}\eta_{ij}^H\,\widetilde{\tau}_j,$$

$$p_{ij}\widetilde{\theta}_i + p_{i+1,j}\,\overline{\theta}_i \le 2\delta_{ij}^H \qquad p_{ij}\widetilde{\theta}_i + p_{i+1,j}\,\overline{\theta}_i \le 2\delta_{ij}^H - \frac{k_j}{h_i}\eta_{ij}^H\,\overline{\tau}_j,$$

$$q_{ij}\widetilde{\tau}_j + q_{i,j+1}\,\overline{\tau}_j \le 2\delta_{ij}^V \qquad q_{ij}\widetilde{\tau}_j + q_{i,j+1}\,\overline{\tau}_j \le 2\delta_{ij}^V + \frac{h_i}{k_j}\zeta_{ij}^V\,\widetilde{\theta}_i,$$

$$q_{ij}\widetilde{\tau}_j + q_{i,j+1}\,\overline{\tau}_j \le 2\delta_{ij}^V \qquad q_{ij}\widetilde{\tau}_j + q_{i,j+1}\,\overline{\tau}_j \le 2\delta_{ij}^V - \frac{h_i}{k_j}\zeta_{ij}^V\,\overline{\theta}_i.$$

As the slopes δ_{ij}^H and δ_{ij}^V are positive, the first four inequalities can be satisfied by sufficiently small values of the parameters $\widetilde{\theta}_i$ and $\overline{\theta}_i$, provided the rhs are positive, i.e.

$$2\delta_{ij}^H + \frac{k_j}{h_i}\,\eta_{ij}^H\,\widetilde{\tau}_j \ge 0 \quad \text{and} \quad 2\delta_{ij}^H - \frac{k_j}{h_i}\eta_{ij}^H\,\overline{\tau}_{j-1} \ge 0.$$

In the same way, the last four inequalities can be satisfied by small values of the parameters $\widetilde{\tau}_j$ and $\overline{\tau}_j$, provided the rhs are positive, i.e.

$$2\delta_{ij}^V + \frac{h_i}{k_j}\,\zeta_{ij}^V\,\widetilde{\theta}_i \ge 0 \quad \text{and} \quad 2\delta_{ij}^V - \frac{h_i}{k_j}\,\zeta_{ij}^V\,\overline{\theta}_{i-1} \ge 0.$$

Again the latter inequalities can be satisfied by taking sufficiently small values of parameters, whatever the signs of quantities η_{ij} and ζ_{ij}.

6.2 Global Biconvexity

Assuming the data are convex in both directions, we obtain, as for bimonotonicity, the following inequalities on the parameters:

$$\begin{aligned}
\zeta^H_{ij}\overline{\theta}_i - \frac{k_j}{h_i}\eta^H_{ij}\widetilde{\tau}_j &\le 2(\delta^H_{ij} - p_{ij}),\\
\zeta^H_{ij}\widetilde{\theta}_i + \frac{k_j}{h_i}\eta^H_{ij}\widetilde{\tau}_j &\le 2(p_{i+1,j} - \delta^H_{ij}),\\
\zeta^H_{i,j+1}\overline{\theta}_i - \frac{k_j}{h_i}\eta^H_{i,j+1}\overline{\tau}_j &\le 2(\delta^H_{i,j+1} - p_{i,j+1}),\\
\zeta^H_{i,j+1}\tilde{\theta}_i - \frac{k_j}{h_i}\eta^H_{i,j+1}\overline{\tau}_j &\le 2(p_{i+1,j+1} - \delta^H_{i,j+1}),\\
-\frac{h_i}{k_j}\zeta^V_{ij}\overline{\theta}_i + \overline{\eta}^V_{ij}\overline{\tau}_j &\le 2(\delta^V_{ij} - q_{ij}),\\
\frac{h_i}{k_j}\zeta^V_{ij}\overline{\theta}_i + \overline{\eta}^V_{ij}\widetilde{\tau}_j &\le 2(q_{i,j+1} - \delta^V_{ij}),\\
\frac{h_i}{k_j}\overline{\zeta}^V_{i+1,j}\overline{\theta}_i + \overline{\eta}^V_{i+1,j}\overline{\tau}_j &\le 2(\delta^V_{i+1,j} - q_{i+1,j}),\\
\frac{h_i}{k_j}\zeta^V_{i+1,j}\overline{\theta}_i + \eta^V_{i+1,j}\widetilde{\tau}_j &\le 2(q_{i+1,j+1} - \delta^V_{i+1,j}).
\end{aligned}$$

As the rhs and the coefficients ζ^H_{ij} and $\bar{\eta}^V_{ij}$ are positive because of the biconvexity of Hermite data, all these inequalities can be satisfied by sufficiently small values of the parameters $\widetilde{\theta}_i, \overline{\theta}_i, \widetilde{\tau}_j$ and $\overline{\tau}_j$.

6.3 Remark on Values of Parameters

When the number of data points is large, each pair of parameters, $(\widetilde{\theta}_i, \overline{\theta}_i)$ or $(\widetilde{\tau}_j, \overline{\tau}_j)$, has to satisfy a large number of inequalities. When the data have strong variations, it may happen that the values of parameters are very small. It causes the appearance of flat regions on the interpolating surface. This is the main drawback of this type of methods based on tensor products.

7 Representation in Terms of Rational B-Splines

The representation of a patch in TP Hermite and Bernstein bases, given in Section 3, is local. In this final section, we show how to derive a TP B-spline basis for the global representation of the rational biquadratic surface.

7.1 Univariate Rational B-Splines

Let $\mathscr{X}_m := \{a = x_0 < x_2 < \ldots < x_{2m} = b\}$ be a partition of $[a,b]$, with the weights $(\widetilde{w}_i, w_i, \overline{w}_i)$ (and the associated parameters $(\widetilde{\theta}_i, \overline{\theta}_i)$) in each subinterval $[x_{2i}, x_{2i+2}]$ of

length $h_i := x_{2i+2} - x_{2i}$. We set $x_{2i+1} := x_{2i} + \frac{1}{2}h_i$. Consider the local Bernstein basis composed of the 4 rational quadratic splines $\{\rho_i^{(k)}(t), k = 0,1,2,3\}$ defined in Section 2.3, the variable t being shifted to $[-1,1]$ through to the change of variable $x = \frac{1}{2}((1-t)x_{2i} + (1+t)x_{2i+2})$. The local space can also be generated by the B-splines $\{\mathscr{M}_{2i}, \mathscr{M}_{2i+1}, \mathscr{M}_{2i+2}, \mathscr{M}_{2i+3}\}$ whose control polygons have the following ordinates

$$(\widetilde{\mu}_i, 0, 0, 0) \qquad (\overline{\mu}_i, 1, 0, 0) \qquad (0, 0, 1, \widetilde{\mu}_{i+1}) \qquad (0, 0, 0, \overline{\mu}_{i+1}),$$

the ordinates at points of the partition being respectively

$$\widetilde{\mu}_i := \frac{h_i \widetilde{\theta}_i}{h_{i-1}\overline{\theta}_{i-1} + h_i \widetilde{\theta}_i}, \quad \overline{\mu}_i := 1 - \widetilde{\mu}_i = \frac{h_{i-1}\overline{\theta}_{i-1}}{h_{i-1}\overline{\theta}_{i-1} + h_i \widetilde{\theta}_i}.$$

From the properties of the local Bernstein basis, one deduces that these B-splines form a partition of unity. Moreover, they are globally C^1 as the left and right derivatives of $\mathscr{M}_{2i}$ and $\mathscr{M}_{2i+1}$ at the point x_{2i} are respectively

$$\mathscr{M}'_{2i}(x_{2i}^-) = \frac{2(\widetilde{\mu}_i - 1)}{h_{i-1}\overline{\theta}_{i-1}} = -\frac{2}{h_{i-1}\overline{\theta}_{i-1} + h_i \widetilde{\theta}_i} = -\frac{2\overline{\mu}_i}{h_{i-1}\overline{\theta}_{i-1}} = -\mathscr{M}'_{2i+1}(x_{2i}^-),$$

$$\mathscr{M}'_{2i}(x_{2i}^+) = -\frac{2\widetilde{\mu}_i}{h_i \widetilde{\theta}_i} = -\frac{2}{h_{i-1}\overline{\theta}_{i-1} + h_i \widetilde{\theta}_i} = -\frac{2(1-\overline{\mu}_i)}{h_i \widetilde{\theta}_i} = -\mathscr{M}'_{2i+1}(x_{2i}^+).$$

Therefore the support of $\mathscr{M}_{2i}$ is the interval $[x_{2i-2}, x_{2i+1}]$, while that of $\mathscr{M}_{2i+1}$ is the interval $[x_{2i-1}, x_{2i+2}]$, for indices $1 \le i \le m-1$. The support of $\mathscr{M}_0$ is $[x_0, x_1]$ and that of $\mathscr{M}_1$ is $[x_0, x_2]$. Likewise, the support of $\mathscr{M}_{2m}$ is $[x_{2m-2}, x_{2m}]$ and that of $\mathscr{M}_{2m+1}$ is $[x_{2m-1}, x_{2m}]$. Thus we get the collection of $2m+2$ B-splines $\{\mathscr{M}_i(x), i \in K_m\}$, with $K_m := \{0 \le i \le 2m+1\}$. A similar construction is valid for the interval $[c,d]$ endowed with the partition $\mathscr{Y}_n := \{c = y_0 < y_2 < \ldots < y_{2n} = d\}$. We then get the collection of $2n+2$ B-splines $\{\mathscr{N}_j(y), j \in L_n\}$ with $L_n := \{0 \le j \le 2n+1\}$, whose local control polygons in the subinterval $[y_{2j}, y_{2j+2}]$ have ordinates

$$(\widetilde{\nu}_j, 0, 0, 0) \qquad (\overline{\nu}_j, 1, 0, 0) \qquad (0, 0, 1, \widetilde{\nu}_{j+1}) \qquad (0, 0, 0, \overline{\nu}_{j+1})$$

the ordinates at points of the partition being respectively

$$\widetilde{\nu}_j := \frac{k_j \widetilde{\tau}_j}{k_{j-1}\overline{\tau}_{j-1} + k_j \widetilde{\tau}_j}, \quad \overline{\nu}_j := 1 - \widetilde{\nu}_j = \frac{k_{j-1}\overline{\tau}_{j-1}}{k_{j-1}\overline{\tau}_{j-1} + k_j \widetilde{\tau}_j}.$$

7.2 Bivariate Rational B-Splines

As usual, the B-spline basis of our bivariate space is defined as the collection of TP B-splines $\{M_\alpha, \alpha = (i,j) \in K_m \times L_n\}$ defined by $M_\alpha(x,y) := \mathscr{M}_i(x)\mathscr{N}_j(y)$, and the Hermite interpolant can be expressed as

$$S(x,y) = \sum_{\alpha \in K_m \times L_n} c_\alpha M_\alpha(x,y).$$

The global *B-spline control net* has vertices $\mathbf{c}_\alpha := (s_\alpha, t_\alpha, c_\alpha) \in \mathbb{R}^3$, whose projections on Ω have the coordinates

$$s_{2i+1} = x_{2i} + \frac{1}{2}\widetilde{\theta}_i h_i, \qquad s_{2i+2} = x_{2i+2} - \frac{1}{2}\overline{\theta}_i h_i,$$
$$t_{2j+1} = y_{2j} + \frac{1}{2}\widetilde{\tau}_j k_j, \qquad t_{2j+2} = y_{2j+2} - \frac{1}{2}\overline{\tau}_j k_j.$$

The value of S at the vertex $A_\gamma = (x_{2i}, y_{2j})$, with $\gamma = (i, j)$, is then given by

$$S(A_\gamma) = \sum_\beta c_\beta M_\beta(A_\gamma) = \sum_{r,s=0,1} c_{2i+r,2j+s} M_{2i+r,2j+s}(A_\gamma)$$

$$= \widetilde{\mu}_i \widetilde{\nu}_j c_{2i,2j} + \widetilde{\mu}_i \overline{\nu}_j c_{2i,2j+1} + \overline{\mu}_i \widetilde{\nu}_j c_{2i+1,2j} + \overline{\mu}_i \overline{\nu}_j c_{2i+1,2j+1}.$$

It is a convex combination of the four neighbouring control vertices $\mathbf{c}_\alpha$, for $\alpha = (2i + r, 2j + s)$ and $r, s = 0, 1$. Likewise, the other vertices of the local control net can be expressed as linear combinations of spline control vertices. Therefore, bimonotonicity and biconvexity conditions could also be expressed in terms of the latter.

Acknowledgement

I thank the referees for their useful comments and remarks which significantly improved the presentation of the paper.

References

1. Carlson, R.E., Fritsch, F.N.: Monotone piecewise bicubic interpolation. SIAM J. Numer. Anal. 22, 386–400 (1985)
2. Costantini, P.: On some recent methods for shape-preserving bivariate interpolation. In: Haussmann, W., Jetter, K. (eds.) Multivariate Interpolation and Approximation. ISNM, vol. 94, pp. 59–68. Birkhäuser-Verlag, Basel (1990)
3. Costantini, P., Fontanella, F.: Shape-preserving bivariate interpolation. SIAM J. Numer. Anal. 27(2), 488–506 (1990)
4. Costantini, P., Manni, C.: A local scheme for bivariate comonotone interpolation. Comput. Aided Geom. Design. 8, 371–391 (1991)
5. Costantini, P., Manni, C.: A bicubic shape-preserving blending scheme. Comput. Aided Geom. Design 13, 307–331 (1996)
6. Costantini, P., Pelosi, F.: Data approximation using shape-preserving parametric surfaces. SIAM J. Numer. Anal. 47, 20–47 (2008)
7. Costantini, P., Pelosi, F., Sampoli, M.L.: Boolean surfaces with shape constraints. Comput. Aided Geom. Design 40, 62–75 (2008)
8. Delbourgo, R.: Shape preserving interpolation to convex data by rational functions with quadratic numerator and linear denominator. IMA J. Numer. Anal. 9, 123–136 (1989)
9. Delbourgo, R., Gregory, J.A.: Shape preserving piecewise rational interpolation. SIAM J. Stat. Sci. Comput. 6, 967–976 (1985)
10. DeVore, R.A., Lorentz, G.G.: Constructive Approximation. Springer, Berlin (1993)
11. Dubuc, S., Merrien, J.-L.: Dyadic Hermite interpolation on a rectangular mesh. Adv. Comput. Math. 10, 343–365 (1999)

12. Fontanella, F.: Shape preserving surface interpolation. In: Chui, C.K., Schumaker, L.L., Utreras, F.I. (eds.) Topics in Multivariate Approximation, pp. 63–78. Academic Press, New-York (1987)
13. Foucher, F.: Bimonotonicity preserving surfaces defined by tensor product of C^1 Merrien subdivision schemes. In: Cohen, A., Merrien, J.-L., Schumaker, L.L. (eds.) Curves and Surfaces, Saint Malo 2002, pp. 149–157. Nashboro Press, Brentwood (2003)
14. Fritsch, F.N., Carlson, R.E.: Monotonicity preserving bicubic interpolation: a progress report. Comput. Aided Geom. Design 2, 117–121 (1985)
15. Gregory, J.A.: Shape preserving rational spline interpolation. In: Graves-Morris, P., Saff, E., Varga, R.S. (eds.) Rational approximation and interpolation, pp. 431–441. Springer, Berlin (1984)
16. Gregory, J.A.: Shape preserving spline interpolation. Comput. Aided Design 18, 53–58 (1986)
17. Gregory, J.A., Delbourgo, R.: Piecewise rational quadratic interpolation to monotonic data. IMA J. Numer. Anal. 2, 123–130 (1982)
18. Lyche, T., Merrien, J.L.: Hermite subdivision with shape constraints on a rectangular mesh. BIT 46, 831–859 (2006)
19. Manni, C., Sablonnière, P.: C^1 comonotone Hermite interpolation via parametric surfaces. In: Daehlen, M., Lyche, T., Schumaker, L.L. (eds.) Mathematical Methods for Curves and Surfaces, pp. 333–342. Vanderbilt University Press, Nashville (1995)
20. Piţul, P., Sablonnière, P.: On a family of univariate rational Bernstein operators. J. Approx. Theory (in press, 2009)
21. Sablonnière, P.: Some properties of C^1-surfaces defined by tensor products of Merrien subdivision schemes. In: Cohen, A., Merrien, J.-L., Schumaker, L.L. (eds.) Curves and Surfaces, Saint Malo 2002, pp. 363–372. Nashboro Press, Brentwood (2003)
22. Sablonnière, P.: Rational Bernstein and spline approximants: a new approach. JJA (Jaén Journal on Approximation) 1, 37–53 (2009)

Tensor Product B-Spline Mesh Generation for Accurate Surface Visualizations in the NIST Digital Library of Mathematical Functions

Bonita Saunders and Qiming Wang

National Institute of Standards and Technology, Gaithersburg MD 20899, USA
bonita.saunders@nist.gov
http://math.nist.gov/~BSaunders

Abstract. We discuss the use of tensor product B-spline meshes to design accurate three dimensional graphs of high level mathematical functions for the National Institute of Standards and Technology (NIST) Digital Library of Mathematical Functions. The graph data is placed inside a web based format such as VRML (Virtual Reality Modeling Language) or X3D (Extensible 3D) to create an interactive visualization environment where users can carefully examine complicated function attributes such as zeros, branch cuts, poles and other singularities. We describe the grid generation technique and its effectiveness for creating clear and informative visualizations.

1 Introduction

One of the most cited publications from the National Institute of Standards and Technology (NIST) is the Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables first published in 1964 [1]. Currently, NIST is completely rewriting the handbook and will release it in both hardcopy and web-based format as the NIST Digital Library of Mathematical Functions (DLMF). The DLMF will include formulas, methods of computation, references, and software information for nearly forty high level, or special, mathematical functions. The website will feature interactive navigation, a mathematical equation search and dynamic interactive visualizations.

This paper discusses our use of mesh generation techniques to facilitate the accurate 3D plotting of functions over irregular, discontinuous, or multiply connected domains for the visualizations. By modifying an algebraic tensor product spline mesh generation algorithm that we originally designed for problems in aerodynamics and solidification theory, we have created boundary/contour fitted computational grids that capture significant function features such as zeros, poles, branch cuts and other singularities.

While the DLMF visualizations are not designed to compete with the on-demand computational and plotting capabilities provided by many well-known computer algebra packages, the motivation for our work stems from some of the inadequacies of such packages. Commercial packages often have many built-in special functions, but their 3D plots are generally over a rectangular Cartesian mesh, leading to poor and misleading graphs. Also, many packages have trouble properly clipping a function, that is, accurately restricting the graphical representation to the points lying within the range

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 385–393, 2010.

of interest. Furthermore, some packages may display the properly clipped graph inside the package, but provide no acceptable way to export the clipped data for use outside the package. This is particularly important if the data needs to be transformed to a web-based format such as X3D (Extensible 3D) or VRML (Virtual Reality Modeling Language).

We show how our mesh generation algorithm eliminates or lessens the severity of many of these problems. We also talk about progress in making the method more adaptive and discuss other possibilities for improvements.

2 Constructing 3D Graphs for Interactive Visualizations in a Digital Library

The first release of the NIST DLMF will consist of thirty eight chapters authored by special function experts throughout the U. S. and abroad. As primary developers of the visualizations that will be an integral part of the DLMF, we have chosen the number and location of graphics for each chapter after consultation with the authors and DLMF editors. To ensure data accuracy, the plot data is being validated by computing each function by at least two different methods, using commercial packages, publicly available codes, or the author's personal codes. However, another concern is plot accuracy, that is, we want to make sure that the displayed plot accurately represents the graph of the function.

Most commercial packages produce very accurate 2D plots. They typically handle discontinuities automatically or fairly easily with special options, and the packages properly cut, or clip, curves so that only points within the desired range appear. Unfortunately, this is often not the case in 3D. In Figure 1, the plot of the Struve function $L_\nu(x)$, rendered using a popular commercial package, illustrates some of the problems we have encountered. We wanted the plot to show the graph for function values less than or equal to 10, but there is a flat area, or shelf, where values greater than 10 have been set to 10. Also, areas near the poles are not resolved very well. Although the shelf-like area may not concern many users, it might be confusing to students and others unfamiliar with the behaviour of the function. Inputting a command to remove the shelf produces a saw-tooth area that can be even more misleading.

Experienced users can input alternative commands to successfully clip the function properly, but we found that this was still not sufficient for our requirements. Since the computation of the function is over a rectangular Cartesian mesh, the figure can look fine when viewed inside the package, but produce a very irregular color map when the data is transformed to another format, such as VRML (Virtual Reality Modeling Language) for viewing on the web. VRML and its successor, X3D (Extensible 3D) are standard 3D file formats for interactive web-based visualizations [12], [13], [14]. Freely available plugins can be downloaded for most platforms. Of course the figure can be improved by using a much larger number of mesh points, but large data files hamper the performance of our visualizations. The rendering problems can be eliminated or decreased in severity by computing the function over a specially designed boundary/contour fitted mesh. While various unstructured techniques such as Voronoi or Delaunay triangulations and quadtree designs might be used to construct such a mesh, a structured

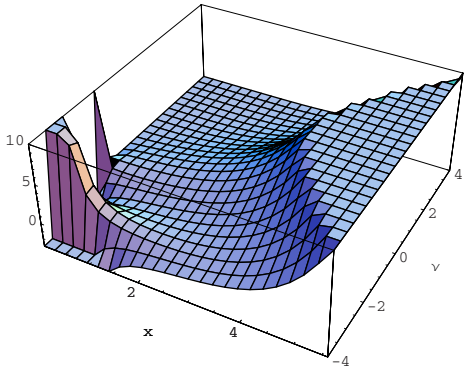


Fig. 1. Plot of Struve function L_v illustrating bad clipping and poor resolution of poles

design allowed us to write more efficient code to implement the interactive capabilities for the visualizations. Whether a structured or unstructured grid is used, it is important that the interior grid lines not stray too far from the contours of the function being plotted. The next section discusses the tensor product B-spline mapping we have used to produce our meshes.

3 Grid Generation Mapping

The complexity of the grid generation problem depends not only on the attributes of the computational domain of the special function, but also on the behaviour of the function. Function domains can range from simple rectangles to complicated multipy connected domains with poles or branch cuts; however, large function gradients can further complicate the creation of a suitable computational grid. Our mesh generation technique is based on an algorithm developed by one of the authors for meshes to be used in solving partial differential equations (pdes) related to aerodynamics and solidification theory [4], [5], [6]. The algorithm constructs a curvilinear coordinate system defined by a mapping $\mathbf{T}$ from the unit square I_2 to a physical domain of arbitrary shape. We let

$$\mathbf{T}(\xi,\eta) = \begin{pmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m}\sum_{j=1}^{n} \alpha_{ij} B_{ij}(\xi,\eta) \\ \sum_{i=1}^{m}\sum_{j=1}^{n} \beta_{ij} B_{ij}(\xi,\eta) \end{pmatrix}, \tag{1}$$

where $0 \le \xi,\eta \le 1$ and each B_{ij} is the tensor product of cubic B-splines. Hence, $B_{ij}(\xi,\eta) = B_i(\xi)B_j(\eta)$ where B_i and B_j are elements of cubic B-spline sequences associated with finite nondecreasing knot sequences, say, $\{s_i\}_1^{m+4}$ and $\{t_j\}_1^{n+4}$, respectively. To obtain the initial α_{ij} and β_{ij} coefficients for $\mathbf{T}$, we first construct the transfinite blending function interpolant that matches the boundary of the physical domain. The interior coefficients are obtained by evaluating the interpolant at knot averages as described in [2] to produce a shape preserving approximation that reproduces straight lines and preserves convexity. For the boundary coefficients, we use the same technique

if the boundary side is a straight line or use DeBoor's SPLINT routine [2] to find coefficients that produce a cubic spline interpolant of the boundary. For simple boundaries, the initial grid is often sufficient, but for more complicated or highly nonconvex boundaries we can improve the grid by using a variational technique to find coefficients that minimize the functional

$$F = \int_{I_2} \left(w_1 \left\{ \left(\frac{\partial J}{\partial \xi} \right)^2 + \left(\frac{\partial J}{\partial \eta} \right)^2 \right\} + w_2 \left\{ \frac{\partial \mathbf{T}}{\partial \xi} \cdot \frac{\partial \mathbf{T}}{\partial \eta} \right\}^2 + w_3 \{uJ^2\} \right) dA \qquad (2)$$

where $\mathbf{T}$ denotes the grid generation mapping, J is the Jacobian of the mapping, w_1, w_2 and w_3 are weight constants, and u represents external criteria for adapting the grid. The integral controls mesh smoothness, orthogonality, and depending on the definition of u, the adaptive concentration of the grid lines. When solving pdes, u might be the gradient of the evolving solution or an approximation of truncation error. Ideally, for our problem, we want u to be based on curvature and gradient information related to the function surface. We have made a slight modification of the integral we gave in [7]. The adaptive term $w_3\{Ju\}$ has been replaced by $w_3\{uJ^2\}$. The new term produces a better concentration of grid lines, and also can be shown to be equivalent to the adaptive term used in the variational technique of Brackbill and Saltzman [3], [8].

To avoid solving the Euler equations for the variational problem, F is approximated in the computer code by the sum

$$\begin{aligned} G = & \sum_{i,j} w_1 \left[\left(\frac{J_{i+1,j} - J_{ij}}{\triangle \xi} \right)^2 + \left(\frac{J_{i,j+1} - J_{ij}}{\triangle \eta} \right)^2 \right] \triangle \xi \triangle \eta \\ & + \sum_{i,j} w_2 Dot_{ij}^2 \triangle \xi \triangle \eta \\ & + \sum_{i,j} w_3 u_{ij} J_{ij}^2 \triangle \xi \triangle \eta \end{aligned} \qquad (3)$$

where J_{ij} is the Jacobian value, u_{ij} is the value of u, and Dot_{ij} is the dot product of $\partial \mathbf{T}/\partial \xi$ and $\partial \mathbf{T}/\partial \eta$ at mesh point (ξ_i, η_j) on the unit square. When $w_3 = 0$, G is actually a fourth degree polynomial in each spline coefficient so the minimum can be found by using a cyclic coordinate descent technique which sequentially finds the minimum with respect to each coefficient. This technique allows the minimization routine to take advantage of the small support of B-splines when evaluating the sums that comprise G.

In Figure 2 we applied the algorithm to a puzzle shaped domain. The initial grid, constructed using linear Lagrange polynomials for the blending functions, is shown on the left. Note that the grid lines overlap the nonconvex boundary. The grid on the right shows the mesh obtained after the spline coefficients are modified to minimize G with weights $w_1 = 1$, $w_2 = 0.5$, and $w_3 = 0$. The overlapping grid lines have been pulled into the interior. When a grid line folds over a boundary, the Jacobian value changes sign. The Jacobian terms of G try to minimize the changes in the Jacobian between adjacent grid cells. Consequently, lines outside the boundary are pulled into the interior. In the original code, this effect was obtained by trying to compute the interval for each coefficient that would guarantee that the Jacobian remained positive [10]. However,

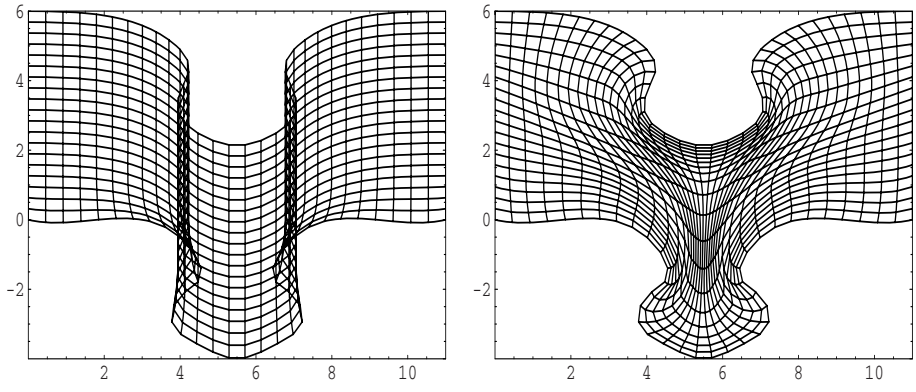


Fig. 2. Initial and optimized puzzle grids

experimenting showed this to be a time consuming procedure that was not needed in most cases. Gonsor and Grandine use the same idea to obtain similar results in [11].

4 Results

We have used boundary/contour fitted grid generation to produce computational grids for over two hundred 3D visualizations for the NIST DLMF. In Figure 3 we show a snapshot of the VRML rendering of the real-valued Struve function $L_\nu(x)$. The data values were obtained by computing the function over the contour grid shown on the left. To create the grid boundary we found the contour, or level, curves where $L_\nu = 10$, the maximum height we wanted to display, and $L_\nu = -3$, the minimum height. We then connected the curves to parts of a rectangle to form a closed boundary for the grid generation algorithm. In contrast to Figure 1, we show that the boundary/contour fitted grid produces a nicely clipped function and very smooth color map. The dark horizontal bar is actually a dense concentration of grid lines needed to accurately resolve the area around a pole. Figure 4 shows a density plot of L_ν that a user may obtain by scaling the surface down so that the vertical component is near zero. This is one of several interactive features that will be available to users of the DLMF.

Figure 5 shows a grid and surface for the modulus of the complex digamma, or psi, function $|\psi(z)|$. The top and bottom halves of the grid were generated separately, with an exponential function used to concentrate the grid points near $y = 0$. The accuracy of the contours satisfying $|\psi(z)| = 6$ is illustrated in the surface shown on the right. Since the coefficients define an explicit curvilinear coordinate system mapping, we can create a coarser, or finer, grid simply by evaluating fewer, or more, points on the unit square. Also, notice that as was seen in Figure 3, the grid spacing does not appear to be smooth in some areas. To guarantee that key boundary or contour points, such as those at zeros or corners, are maintained regardless of grid size, we identify "fixed points," that is, grid points that must always be kept. Grid lines are always drawn through these points. In most cases, the resulting discontinuities in cell spacing do not affect the quality of the

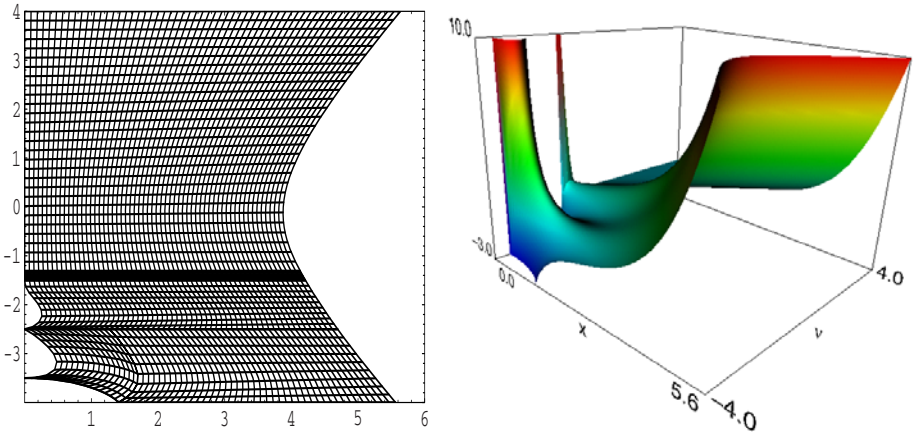


Fig. 3. Grid and Plot of Struve Function L_ν

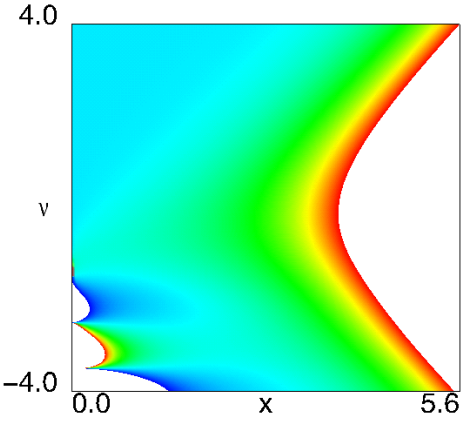


Fig. 4. Density Plot of Struve Function L_ν

visualizations significantly, but this is an aspect of the technique that might be improved by carefully chosen adaptive weights.

All the visualizations in the NIST DLMF represent either real-valued or complex-valued functions of the form, $w = f(x,y)$. For complex-valued functions, the user has the option of using a height based color mapping where height = $|w|$, or a mapping based on the phase, or argument, of w. Figure 6 shows a plot of the modulus of the Hankel function $H_1^{(1)}(z)$ with a phase based color map. On the left is a phase density plot obtained by scaling the figure down in the vertical direction. The branch cut, zeros and pole are visible in both figures.

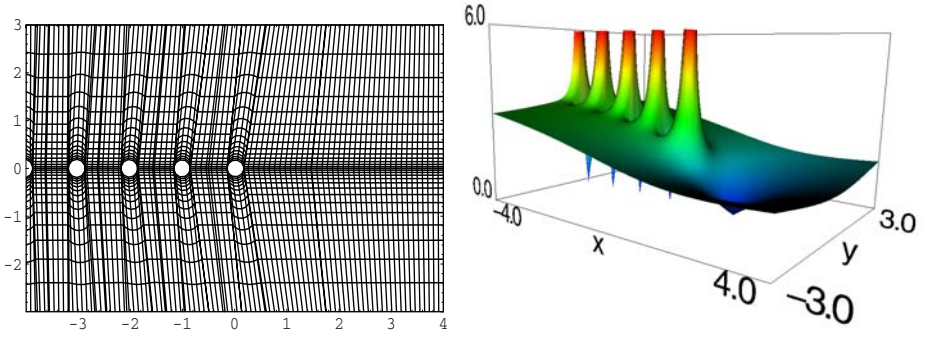


Fig. 5. Grid and Plot of Complex Digamma Function $|\psi(z)|$

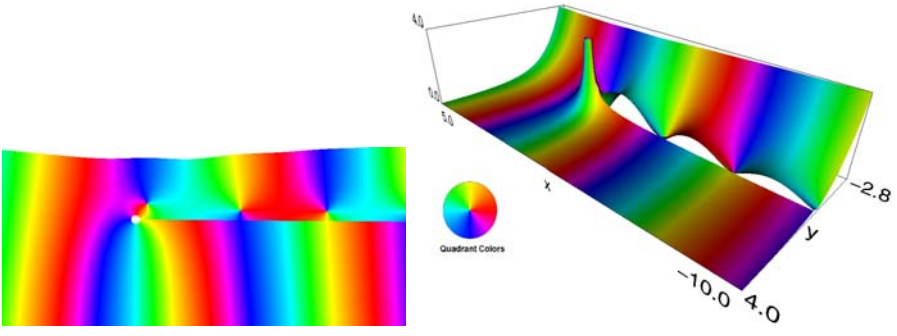


Fig. 6. Plot of Hankel function $H_1^{(1)}$ with phase colormap

In Figure 7 we attracted grid points in an equally spaced square grid to a circle and a sine curve, respectively. To obtain the attraction to the circle we defined the u in the adaptive term of our functional by

$$u(\xi,\eta) = e^{-500[(\xi-.5)^2+(\eta-.5)^2-\frac{9}{64}]^2} \tag{4}$$

and for attraction to the sine curve we defined

$$u(\xi,\eta) = e^{-100[0.35(\sin(2\pi\xi)+1.35)-\eta]^2}. \tag{5}$$

The parameters were chosen so that the curves lie completely within the unit square. Of course these definitions guarantee that the grid lines are attracted to a circle (or sine curve) in I_2 but not in the xy plane. Choosing u in this manner allowed us to quickly test the affect of u with few modifications to our code since the functional approximation G will still be a fourth degree polynomial in each spline coefficient. To show the affect of u we simply mapped the unit square to a larger square.

In order to reach our goal of adapting the grid lines to function curvature data, we must update the code to allow u to have a nonlinear dependence on the spline coefficients. We have made some progress in this direction by incorporating Nash's truncated

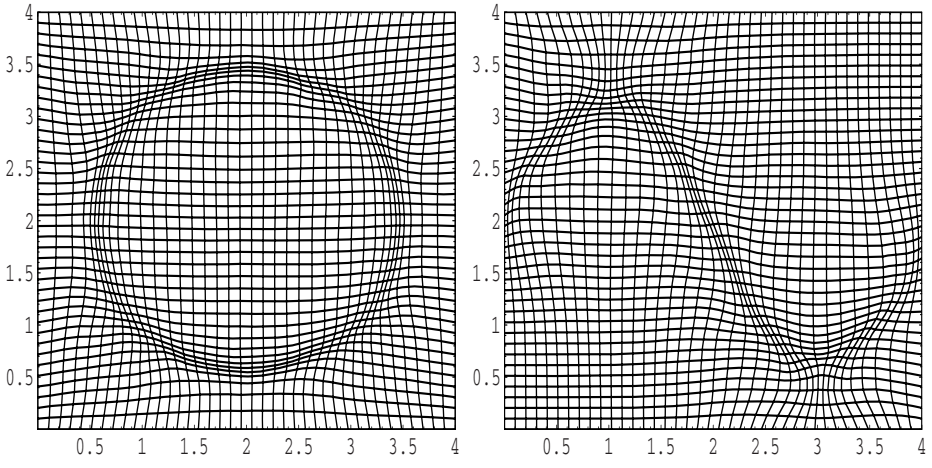


Fig. 7. Grids adapted to circular and sinusoidal shapes

Newton algorithm code described in [9], but additional modifications to take better advantage of the small support of B-splines are needed. We also need to look at the movement of the boundary points. Minimization of G should also include adjustment of the boundary coefficients, but this must be done very carefully so that we obtain a reparameterization of the boundary curves without altering the shape of the boundary. Then we can concentrate on defining an expression for u that captures the appropriate function gradient information. We are acutely aware that the specialized nature of some high level mathematical functions may mean that accessing and linking the codes needed to compute gradient and curvature data may not be a trivial task.

5 Conclusions

We have used boundary/contour fitted grid generation to complete over two hundred interactive 3D visualizations for the NIST Digital Library of Mathematical Functions. Our tensor product B-spline technique has been effective in addressing many problems that appear in commercial packages such as the inaccurate resolution of poles, bad clipping, and poor color mappings. The clarity of the surface color map is still an issue in areas where the gradient is large, but we believe this can be improved by using an adaptive technique based on function gradient and curvature information. The adaptive method should also result in smaller data file sizes, which can improve the efficiency of some of our interactive features.

In spite of tackling several issues simultaneously, including the production of the visualizations, data validation, the availability of VRML/X3D plugins, and accessibility on major platforms, we have made steady progress toward the development of the adaptive code. We have successfully adapted our grids to curves defined on the unit square and had some success with adaption to more general curves where the definition is influenced by the spline coefficients.

Our future work will include incorporating the movement of the boundary coefficients in the minimization functional, describing suitable adaption criteria that will capture gradient and curvature information, and integrating the computation of this function information with the grid generation code.

Disclaimer

All references to commercial products are provided only for clarification of the results presented. Their identification does not imply recommendation or endorsement by NIST.

References

1. Abramowitz, M., Stegun, I.A. (eds.): Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables. National Bureau of Standards Applied Mathematics Series, vol. 55. U.S. Government Printing Office, Washington (1964)
2. de Boor, C.: A Practical Guide to Splines, Revised Edition. Springer, New York (2001)
3. Brackbill, J.U., Saltzman, J.S.: Adaptive zoning for singular problems in two dimensions. J. Comput. Phys. 46, 342–368 (1982)
4. Saunders, B.V.: A boundary conforming grid generation system for interface tracking. Computers Math. Applic. 29, 1–17 (1995)
5. Saunders, B.V.: The application of numerical grid generation to problems in computational fluid dynamics. Council for African American Researchers in the Mathematical Sciences, Contemporary Mathematical Series, vol. III, 275. American Mathematical Society (2001)
6. Saunders, B.V., Wang, Q.: From 2d to 3d: numerical grid generation and the visualization of complex surfaces. In: Soni, B.K., et al. (eds.) Proceedings of the 7th International Conference on Numerical Grid Generation in Computational Field Simulations, Whistler, British Columbia, Canada, September 25-28 (2000)
7. Saunders, B.V., Wang, Q.: From B-spline mesh generation to effective visualizations for the nist digital library of mathematical functions. In: Chenin, P., et al. (eds.) Curve and Surface Design: Avignon 2006. Nashboro Press, Brentwood (2007)
8. Thompson, J.F., Warsi, Z.U.A., Mastin, C.W.: Numerical Grid Generation: Foundations and Applications. North Holland, New York (1985)
9. Nash, S.G.: Newton-type minimization via the lanczos method. SIAM Journal on Numerical Analysis 21(4), 770–788 (1984)
10. Saunders, B.V.: Algebraic grid generation using tensor product B-splines, NASA CR-177968 (1985)
11. Gonsor, D., Grandine, T.: A curve blending algorithm suitable for grid generation. In: Lucian, M., et al. (eds.) Geometric Modeling and Computing: Seattle 2003. Nashboro Press, Brentwood (2004)
12. Carey, R., Bell, G.: The Annotated VRML 2.0 Reference Manual. Addison-Wesley, Boston (1997)
13. Brutzman, D., Daly, L.: Extensible 3D Graphics for WEB Authors. Morgan Kaufmann (Elsevier), San Francisco (2007)
14. VRML. The Virtual Reality Modeling Language, International Standard ISO/IEC 14772-1:1997 (1997)

Low Degree Euclidean and Minkowski Pythagorean Hodograph Curves

Zbyněk Šír[1] and Jiří Kosinka[2]

[1] Faculty of Mathematics and Physics, Charles University in Prague,
Sokolovská 83, 186 75 Prague
zbynek.sir@mff.cuni.cz
[2] Centre of Mathematics for Applications, Department of Informatics, University of Oslo,
P.O. Box 1053, Blindern, 0316 Oslo, Norway
jiri.kosinka@email.cz

Abstract. In our contribution we study cubic and quintic Pythagorean Hodograph (PH) curves in the Euclidean and Minkowski planes. We analyze their control polygons and give necessary and sufficient conditions for cubic and quintic curves to be PH. In the case of Euclidean cubics the conditions are known and we provide a new proof. For the case of Minkowski cubics we formulate and prove a new simple geometrical condition. We also give conditions for the control polygons of quintics in both types of planes.

Moreover, we introduce the new notion of the *preimage of a transformation*, which is closely connected to the so-called preimage of a PH curve. We determine which transformations of the preimage curves produce similarities of PH curves in both Euclidean and Minkowski plane. Using these preimages of transformations we provide simple proofs of the known facts that up to similarities there exists only one Euclidean PH cubic (the so-called Tschirnhausen cubic) and two Minkowski PH cubics. Eventually, with the help of this novel approach we classify and describe the systems of Euclidean and Minkowski PH quintics.

1 Introduction

Pythagorean Hodograph (PH) curves (see the survey [15], the book [17] and the references cited therein) form an interesting subclass of polynomial parametric curves. They possess a piecewise polynomial arc length function and rational offset curves in the planar case. PH curves provide an elegant solution to various difficult problems occurring in many applications, in particular in the context of CNC (computer-numerical-control) machining and offset approximations by rational splines. The notion of PH curves was also generalized to Minkowski plane and space with applications to offsets trimming and medial axis transform approximations.

Since their introduction in [10] planar PH curves have been studied in many publications. The greatest attention was devoted to various construction techniques [2, 9, 12, 13, 20, 22, 28, 30]. The number of publications devoted to structural and classification results is much smaller. The original publication [10] and later [1] studying PH curves from the point of view of complex numbers seem to be the most important. In [34] the conditions for the monotonicity of curvature were studied for PH quintics.

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 394–418, 2010.

Spatial PH curves were introduced in [11], and they have later been characterized using results about Pythagorean quadruples in the ring of polynomials and quaternion calculus [7,14]. Various construction and other techniques are also available [4,16,18, 21,32,33].

In the Minkowski setting, Minkowski Pythagorean hodograph (MPH) curves were introduced by [29]. They are important in the context of the medial axis transform and approximation of planar domain boundaries and their offsets. MPH curves and their applications were studied in [5,24,25,32]. In [23] a classification of planar and spatial MPH cubics was given. The importance of planar Minkowski curves consist in the fact that they (when embedded in the Minkowski space) describe certain symmetric domains.

In [6] a unified approach to PH and MPH curves in planar and spatial settings based on Clifford algebra calculus was proposed.

In this paper we analyze the types and properties of control polygons of PH cubics and quintics in Euclidean and Minkowski planes. The case of Euclidean cubics was thoroughly treated in [10]. We give new proofs of some of the results of this paper and provide an analogical analysis for PH cubics in the Minkowski plane and planar PH and MPH quintics. Moreover, we introduce a novel approach to classifying PH curves using the so-called *preimage of a transformation*. This notion is closely related to the preimage curve of a PH curve. Based on this new approach, as the main result of the present paper, we give a classification of planar PH and MPH quintics and give simpler proofs of the classification of planar and spatial PH and MPH cubics, cf. [10,11,23,32].

The paper is organized as follows. In Section 2 we recall some basic facts about the Minkowski plane, Clifford algebras and PH curves. Section 3 introduces the preimage of a transformation, a notion which plays a key role in classifying PH curves. In Section 4 we study the control polygon properties and shapes of cubics in the Euclidean and Minkowski planes. In Section 5 we provide a similar analysis for Euclidean and Minkowski PH quintics. Finally we conclude the paper.

2 Preliminaries

In this section we summarize some fundamental results concerning PH curves in the Euclidean and Minkowski plane and their representation using complex numbers and Clifford algebra $\mathscr{C}\ell(1,1)$.

2.1 Minkowski Plane

The two–dimensional Minkowski space $\mathbb{R}^{1,1}$ is a real linear space with an indefinite inner product given by the matrix $G = \mathrm{diag}(1,-1)$. We use the upper index $1,1$ to emphasize the signature of the inner product. Some authors use $\mathbb{R}^2$ or $\mathbb{R}^{2,1}$ to denote the Minkowski plane.

The inner product of two vectors $\mathbf{u} = (u_1,u_2)^\top$, $\mathbf{v} = (v_1,v_2)^\top$, $\mathbf{u},\mathbf{v} \in \mathbb{R}^{1,1}$ is defined as

$$\langle \mathbf{u},\mathbf{v} \rangle = \mathbf{u}^\top G \mathbf{v} = u_1 v_1 - u_2 v_2. \tag{1}$$

The two axes spanned by the vectors $\mathbf{e}_1 = (1,0)^\top$ and $\mathbf{e}_2 = (0,1)^\top$ will be denoted as the x– and y–axis, respectively.

Since the quadratic form defined by G is not positive definite as in the Euclidean case, the square norm of $\mathbf{u}$ defined by $||\mathbf{u}||^2 = \langle \mathbf{u}, \mathbf{u} \rangle$ may be positive, negative or zero. Motivated by the theory of relativity one distinguishes three so-called 'causal characters' of vectors. A vector $\mathbf{u}$ is said to be space–like if $||\mathbf{u}||^2 > 0$, time–like if $||\mathbf{u}||^2 < 0$, and light–like (or isotropic) if $||\mathbf{u}||^2 = 0$.

A linear mapping $L : \mathbb{R}^{1,1} \to \mathbb{R}^{1,1}$ is called a *Lorentz transform* or Minkowski rotation if it preserves the Minkowski inner product, i.e. $\langle \mathbf{u}, \mathbf{v} \rangle = \langle L\mathbf{u}, L\mathbf{v} \rangle$ for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{1,1}$. The Lorentz transforms form the *Lorentz group* $\mathscr{L} = O(1,1)$.

Any Lorentz transform is described by a 2×2–matrix $L = (l_{i,j})_{i,j=1,2}$. Its column vectors $\mathbf{l}_1$ and $\mathbf{l}_2$ satisfy $\langle \mathbf{l}_i, \mathbf{l}_j \rangle = G_{i,j}$, $i, j \in \{1,2\}$, i.e., they form an orthonormal basis of $\mathbb{R}^{1,1}$ with respect to the inner product (1). The equation $\langle \mathbf{l}_2, \mathbf{l}_2 \rangle = G_{2,2} = -1$ implies $l_{22}^2 \geq 1$. The Lorentz transform L is said to be *orthochronous* if $l_{22} \geq 1$. Obviously, the determinant of any Lorentz transform L equals to ± 1. The *special* ones are characterized by $\det(L) = 1$.

The Lorentz group $\mathscr{L}$ consists of four components. The special orthochronous Lorentz transforms form the subgroup $SO_+(1,1)$ which consists of matrices of the type

$$\begin{pmatrix} \cosh\alpha & \sinh\alpha \\ \sinh\alpha & \cosh\alpha \end{pmatrix}.$$

The remaining three components are $T_1 SO_+(1,1)$, $T_2 SO_+(1,1)$ and $T_1 T_2 SO_+(1,1)$, where $T_1 = \mathrm{diag}(1,-1)$ and $T_2 = \mathrm{diag}(-1,1)$.

2.2 The Clifford Algebra $\mathscr{C}\ell(1,1)$

Any real linear space, which is equipped with a non–degenerate quadratic form, has an associated Clifford algebra, see [6,27] for a more detailed introduction. In particular we are interested in the Clifford algebra $\mathscr{C}\ell(1,1)$, which is linked to the Minkowski plane $\mathbb{R}^{1,1}$.

This Clifford algebra has three different classes of basis elements: the *scalar* identity element $\mathbf{1}$, the orthonormal basis *vectors* $\mathbf{e}_1$, $\mathbf{e}_2$ and the *pseudo–scalar* $\mathbf{e}_{12}$. The non–commutative multiplication $\cdot$ can be deduced from the basic relations $\mathbf{e}_1^2 = -\mathbf{e}_2^2 = \mathbf{1}$ and $\mathbf{e}_1 \cdot \mathbf{e}_2 = \mathbf{e}_2 \cdot \mathbf{e}_1 \mathbf{e}_{12}$, see Table 1.

Any element of the Clifford algebra is a linear combination of these basis elements. In order to simplify the notation, we shall use vectors in $\mathbb{R}^4$ to represent them,

$$A = [a_0, a_1, a_2, a_3] = a_0\mathbf{1} + a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_{12}.$$

Table 1. Multiplication table for $\mathscr{C}\ell(1,1)$

$\cdot$	$\mathbf{1}$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_{12}$
$\mathbf{1}$	$\mathbf{1}$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_{12}$
$\mathbf{e}_1$	$\mathbf{e}_1$	$\mathbf{1}$	$\mathbf{e}_{12}$	$\mathbf{e}_2$
$\mathbf{e}_2$	$\mathbf{e}_2$	$-\mathbf{e}_{12}$	$-\mathbf{1}$	$-\mathbf{e}_1$
$\mathbf{e}_{12}$	$\mathbf{e}_{12}$	$-\mathbf{e}_2$	$\mathbf{e}_1$	$\mathbf{1}$

The *conjugation* of elements and the *squared norm* are defined as

$$\bar{A} = [a_0, -a_1, -a_2, -a_3] \quad \text{and}$$
$$N(A) = A \cdot \bar{A} = a_0^2 - a_1^2 + a_2^2 - a_3^2$$

respectively. The operation of conjugation satisfies $\overline{A \cdot B} = \bar{B} \cdot \bar{A}$.

All vectors of $\mathbb{R}^{1,1}$ will be identified with *pure vectors*

$$\mathbf{c} = (c_1, c_2)^\top = c_1\mathbf{e}_1 + c_2\mathbf{e}_2 = [0, c_1, c_2, 0] \in \mathscr{C}\ell(1,1). \tag{2}$$

The norms in $\mathbb{R}^{1,1}$ and $\mathscr{C}\ell(1,1)$ are related by

$$||\mathbf{c}||^2 = -N(\mathbf{c}).$$

The set of *scalars* combined with *pseudoscalars* forms a subalgebra

$$\mathbb{H}^{1,1} = \mathbb{R}\mathbf{1} + \mathbb{R}\mathbf{e}_{12}$$

of the Clifford algebra $\mathscr{C}\ell(1,1)$. Its elements will be represented by *calligraphic* characters, $\mathscr{A} = [a_0, 0, 0, a_3] = a_0\mathbf{1} + a_3\mathbf{e}_{12}$.

Definition 1. *We define on the algebra $\mathscr{C}\ell(1,1)$ the commutative multiplication*

$$A \star B := \frac{1}{2}(A \cdot \mathbf{e}_1 \cdot \bar{B} + B \cdot \mathbf{e}_1 \cdot \bar{A}).$$

Corresponding n-th powers will be denoted $A^{n\star} = \underbrace{A \star A \star \ldots \star A}_{n\times}$.

Note, that $\mathscr{A} \star \mathscr{B}$ is always a pure vector.

Now we state the following Lemmas, which we will make use of in the next section.

Lemma 2. *Let* $\mathbf{a}$ *and* $\mathbf{b}$ *be two pure vectors. Then*

$$N(\mathbf{a} \star \mathbf{b}) = -N(\mathbf{a})N(\mathbf{b}). \tag{3}$$

Proof. By using the definitions of Clifford norm and Minkowski inner product, one can check that both sides of (3) are equal to $-(a_1^2 - a_2^2)(b_1^2 - b_2^2)$. □

Lemma 3. *Let* $\mathbf{h} = [0, h_1, h_2, 0]$ *be a pure space–like or light–like vector and* $\mathscr{A} = [a_0, 0, 0, a_3] \in \mathbb{H}^{1,1}$. *Then all solutions of the equation*

$$\mathbf{h} = \mathscr{A} \star \mathscr{A} \tag{4}$$

are given by

$$a_0 = \pm\sqrt{\frac{h_1 \pm \sqrt{h_1^2 - h_2^2}}{2}}, \quad a_3 = \mp\sqrt{\frac{h_1 \mp \sqrt{h_1^2 - h_2^2}}{2}}.$$

Proof. By using the definition of the symmetric product, the equation (4) is equivalent to the system

$$a_0^2 + a_3^2 = h_1 \quad \text{and} \quad 2a_0a_3 = -h_2.$$

Solving this system, which yields a biquadratic equation, concludes the proof. □

2.3 Pythagorean Hodograph Curves

Given any vector space $\mathbb{V}$ with (possibly indefinite) dot product we can define Pythagorean hodograph (PH) curves, as polynomial curves in $\mathbb{V}$ with polynomial speed, i.e., as polynomial mappings $\mathbf{p} : \mathbb{R} \to \mathbb{V}$ such that $||\mathbf{p}'||$ is a polynomial in t. Obviously, this definition is invariant with respect to the mappings preserving the dot product. In this paper we will analyse in detail the cases $\mathbb{V} = \mathbb{R}^2$ and $\mathbb{V} = \mathbb{R}^{1,1}$. However, many of our approaches can be applied in general.

2.4 Planar Euclidean PH Curves

The *hodograph* of a planar curve $\mathbf{p}(t) = (x(t), y(t))^\top$ of degree n is the curve $\mathbf{h}(t) = (x'(t), y'(t))^\top$ of degree $n-1$, where $'$ denotes the first derivative. For the remainder of the paper the hodograph of a curve $\mathbf{p}(t)$ will be denoted $\mathbf{h}(t)$.

Recall that a polynomial curve is called *Pythagorean Hodograph (PH)*, see [10] if the length of its tangent vector depends in a (piecewise) polynomial way on the parameter. More precisely, $\mathbf{p}(t) = (x(t), y(t))^\top$ is called a *planar PH curve* if there exists a polynomial $\sigma(t)$ such that

$$x'(t)^2 + y'(t)^2 = \sigma^2(t). \tag{5}$$

Following the classical result of [26] the general solution of (5) takes the form

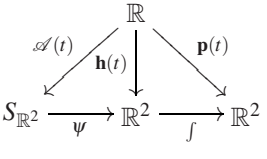
$$\begin{aligned} x'(t) &= q(t)(u^2(t) - v^2(t)), \\ y'(t) &= q(t)(2u(t)v(t)), \\ \sigma(t) &= q(t)(u^2(t) + v^2(t)). \end{aligned}$$

If all real roots of $\gcd(x'(t), y'(t))$ have even multiplicity[1], then $q(t)$ can be omitted. In the sequel we will suppose this case, since the common factor $q(t)$ raises the degree of the curve without yielding more geometric flexibility (it does not modify tangent directions).

Any PH curve can be therefore represented through its preimage

$$\mathscr{A}(t) = (u(t), v(t)).$$

We will call the two dimensional vector space of u and v the representation space and denote it $S_{\mathbb{R}^2}$. Then we can describe the whole situation by the following diagram



where $\psi : (u, v)^\top \to (u^2 - v^2, 2uv)^\top$ will be called *representation mapping*.

The representation result can be reformulated using complex numbers. Any planar curve $\mathbf{p}(t) = (x(t), y(t))^\top$ is identified with the complex function $\mathbf{p}(t) = x(t) + y(t)\mathbf{i}$

[1] This includes the generic case $\gcd(x'(t), y'(t)) = 1$.

and any preimage $\mathscr{A} = (u(t), v(t))^\top$ with the complex function $\mathscr{A}(t) = u(t) + v(t)\mathbf{i}$. Then the characterization mapping ψ corresponds to complex squaring, since $\mathscr{A}^2 = (u + v\mathbf{i})^2 = (u^2 - v^2) + 2uv\mathbf{i}$. The relation between the hodograph of a PH curve and its preimage thus becomes

$$\mathbf{h}(t) = \mathscr{A}(t)^2.$$

2.5 Minkowski Pythagorean Hodograph Curves

Following [29], *Minkowski Pythagorean Hodograph* (MPH) curves are defined similarly to PH curves, but with respect to the norm induced by the Minkowski inner product. More precisely, a polynomial curve $\mathbf{p} \in \mathbb{R}^{1,1}$, $\mathbf{p}(t) = (x(t), y(t))^\top$ is called an MPH curve if

$$||\mathbf{h}(t)||^2 = x'^2(t) - y'^2(t) = \sigma^2(t) \tag{6}$$

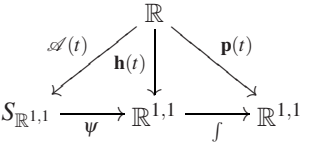
for some polynomial σ.

Due to the definition of MPH curves, the tangent vector $\mathbf{p}'(t)$ cannot be time-like. Also, light-like tangent vectors correspond to roots of the polynomial $\sigma(t)$ in (6). Each regular point of an MPH curve has a space-like or light-like tangent vector.

The equation (6) holds if and only if there exist polynomials $u(t)$, $v(t)$, $q(t)$ such that

$$\begin{aligned} x'(t) &= q(t)(u^2(t) + v^2(t)), \\ y'(t) &= q(t)(2u(t)v(t)), \\ \sigma(t) &= q(t)(u^2(t) - v^2(t)). \end{aligned}$$

Again we will suppose that $q(t) = 1$, in which case the following representation diagram holds



where the representation mapping is $\psi : (u, v)^\top \to (u^2 + v^2, 2uv)^\top$.

The representation result can be reformulated using Clifford algebra $\mathscr{C}\ell(1,1)$, see [6], by identifying vectors with elements of the algebra, cf. (2). Any planar curve $\mathbf{p}(t) = (x(t), y(t))^\top$ is identified with the function $\mathbf{p}(t) = x(t)\mathbf{e}_1 + y(t)\mathbf{e}_2$ and any preimage $\mathscr{A} = (u(t), v(t))^\top$ with the function in the subalgebra $\mathbb{H}^{1,1}$ by setting $\mathscr{A}(t) = u(t) + v(t)\mathbf{e}_{12}$. Then the characterization mapping ψ corresponds to $\mathscr{A}(t) \cdot \mathbf{e}_1 \cdot \bar{\mathscr{A}}(t)$ and the relation between a PH curve and its preimage thus becomes

$$\mathbf{h}(t) = \mathscr{A}(t) \cdot \mathbf{e}_1 \cdot \bar{\mathscr{A}}(t) = \mathscr{A}(t)^{2\star}.$$

3 Inducing Transformations via Preimages

In the sequel we want to classify low degree PH curves up to "similarities". More precisely, we want to classify PH curves in $\mathbb{V}$ with respect to the action of the conformal

group $CO(\mathbb{V}) = ON(\mathbb{V}) \times \mathbb{R}_+$, where $ON(\mathbb{V})$ is the group of all transformations preserving the inner product. Since the PH curves are represented by preimages in the representation space, we will try to realize the classification already on the level of preimages. In this chapter we will investigate which transformations of the representation space produce transformations of the space $\mathbb{V}$ belonging to $CO(\mathbb{V})$.

3.1 General Concept

As we have seen in Section 2, the general representation scheme for PH curves is

$$S_{\mathbb{V}} \xrightarrow{\ \psi\ } \mathbb{V} \xrightarrow{\ \int\ } \mathbb{V}.$$

The integration commutes with any affine transformation of the hodograph and from the transformation point of view it only can produce a translation through a choice of integration constants. We are looking for certain preimages of conformal mappings, i.e., for linear mappings of $S_{\mathbb{V}}$ which produce conformal mappings of $\mathbb{V}$. This concept is formalized in the following definition.

Definition 4. *An element of $\lambda \in GL(S_{\mathbb{V}})$ will be called the conformal preimage of $\mu \in CO(\mathbb{V})$ if and only if $\psi(\lambda(a)) = \mu(\psi(a))$ for any element $a \in S_{\mathbb{V}}$.*

In other words, the following commutative diagram holds

$$\begin{array}{ccc} S_{\mathbb{V}} & \xrightarrow{\ \psi\ } & \mathbb{V} \\ {\scriptstyle\lambda}\downarrow & & \downarrow{\scriptstyle\mu} \\ S_{\mathbb{V}} & \xrightarrow{\ \psi\ } & \mathbb{V} \end{array}$$

Indeed, one μ can have several preimages. On the other hand, due to the surjectivity of ψ any λ can be the preimage of at most one μ.

Lemma 5. *The set of all preimages of all $\mu \in CO(\mathbb{V})$ forms a subgroup of $GL(S_{\mathbb{V}})$, which will be called the preimage group and denoted $PCO(S_{\mathbb{V}})$.*

Proof. It is a straightforward observation that $\mathbf{1}_{S_{\mathbb{V}}}$ is the preimage of $\mathbf{1}_{\mathbb{V}}$. Moreover, if λ is the preimage of μ and λ' is the preimage of μ' then λ^{-1} is the preimage of μ^{-1} and $\lambda(\lambda')$ is the preimage of $\mu(\mu')$. □

Before we turn our attention to concrete examples of $\mathbb{V}$, we consider the influence of the linear reparameterization of the preimage on the shape of the final PH curve. Indeed, reparameterization is (besides the application of a transformation) another natural way how to modify the preimage. The following lemma follows directly from the integration by substitution rule.

Lemma 6. *Let $\mathscr{A}(t) : \mathbb{R} \to S_{\mathbb{V}}$ be a polynomial preimage of the PH curve*

$$\mathbf{p}(t) = \int \psi(\mathscr{A}(t)) : \mathbb{R} \to \mathbb{V}$$

and $k \neq 0, l$ real constants. Then $\mathscr{A}(kt+l)$ is the preimage of the curve $\frac{1}{k}\mathbf{p}(kt+l)$. Therefore, reparameterizing the preimage corresponds to scaling $\mathbf{p}(t)$.

3.2 Group $PCO(S_{\mathbb{R}^2})$

Recall that in the case of Euclidean planar PH curves, the representation mapping has the form

$$\psi : (u,v)^\top \to (u^2 - v^2, 2uv)^\top.$$

The conformal group $CO(\mathbb{R}^2)$ is generated by mappings of the form $\begin{pmatrix} a & b \\ -b & a \end{pmatrix}$ together with the mapping given by $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. An element $\begin{pmatrix} k & l \\ m & n \end{pmatrix} \in GL(S_{\mathbb{R}^2})$ is the preimage of $\begin{pmatrix} a & b \\ -b & a \end{pmatrix} \in CO(\mathbb{R}^2)$ if and only if for any $(u,v)^\top \in S_{\mathbb{R}^2}$ it holds

$$\begin{pmatrix} a & b \\ -b & a \end{pmatrix} \begin{pmatrix} u^2 - v^2 \\ 2uv \end{pmatrix} = \begin{pmatrix} (ku+lv)^2 - (mu+nv)^2 \\ 2(ku+lv)(mu+nv) \end{pmatrix}.$$

Comparing the coefficients we obtain the following system of equations

$$\begin{aligned} a &= k^2 - m^2 = n^2 - l^2 = kn + lm, \\ b &= kl - mn = -2km = 2ln, \end{aligned}$$

which implies $n = k, m = -l, a = k^2 - l^2, b = 2kl$. These equations are well known in the context of PH curves and it has a solution for any a, b. Similarly,

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \in GL(S_{\mathbb{R}^2})$$

are the two preimages of $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \in GL(\mathbb{R}^2)$. Summing up, we obtain the following result.

Theorem 7. *The preimage group $PCO(S_{\mathbb{R}^2})$ is the conformal group $CO(\mathbb{R}^2)$ and any element of $CO(\mathbb{R}^2)$ has its preimage in $PCO(S_{\mathbb{R}^2})$.*

3.3 Group $PCO(S_{\mathbb{R}^{1,1}})$

Recall that in the case of Minkowski planar PH curves, the representation mapping has the form

$$\psi : (u,v)^\top \to (u^2 + v^2, 2uv)^\top.$$

The conformal group $CO(\mathbb{R}^{1,1})$ is generated by mappings of the form $\begin{pmatrix} a & b \\ b & a \end{pmatrix}$, where

$$|a| > |b|, a > 0 \tag{7}$$

together with the mappings given by $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$.

An element $\begin{pmatrix} k & l \\ m & n \end{pmatrix} \in GL(S_{\mathbb{R}^{1,1}})$ is the preimage of $\begin{pmatrix} a & b \\ b & a \end{pmatrix} \in CO(\mathbb{R}^{1,1})$ if and only if for any $(u,v)^\top \in S_{\mathbb{R}^{1,1}}$ it holds

$$\begin{pmatrix} a & b \\ b & a \end{pmatrix} \begin{pmatrix} u^2+v^2 \\ 2uv \end{pmatrix} = \begin{pmatrix} (ku+lv)^2+(mu+nv)^2 \\ 2(ku+lv)(mu+nv) \end{pmatrix}.$$

Comparing the coefficients we obtain the following system of equations

$$\begin{aligned} a &= k^2+m^2 = n^2+l^2 = kn+lm, \\ b &= kl+mn = 2km = 2ln, \end{aligned}$$

which implies $n=k, m=l, a=k^2+l^2, b=2kl$. Again, when (7) hold, this system has always a solution. Moreover, $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \in GL(S_{\mathbb{R}^{1,1}})$ are the two preimages of $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \in GL(\mathbb{R}^{1,1})$. On the other hand, the element $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \in GL(\mathbb{R}^{1,1})$ has no preimage. Summing up, we arrive at the following result

Theorem 8. *The preimage group $PCO(S_{\mathbb{R}^{1,1}})$ is generated by the elements of the form $\begin{pmatrix} k & l \\ l & k \end{pmatrix}$, where $k \neq l$ can be arbitrary and by the elements $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$. A preimage exists for an element of $CO(\mathbb{R}^{1,1})$ provided that its first matrix element is positive.*

4 Planar PH and MPH Cubics

4.1 Euclidean PH Cubics

We start with recalling the properties of the control polygons of planar PH cubics, cf. [10]. Let $\mathbf{p}(t)$ be a planar cubic in $\mathbb{R}^2$ expressed in the Bernstein-Bézier representation [19]

$$\mathbf{p}(t) = \sum_{i=0}^{3} \mathbf{p}_i B_i^3(t), \quad t \in [0,1],$$

where $\mathbf{p}_i$, are the control points and $B_j^n(t) = \binom{n}{j} t^j (1-t)^{n-j}$ are the Bernstein polynomials. Moreover, let L_1, L_2 and L_3 be the lengths of the control polygon legs and θ_1, θ_2

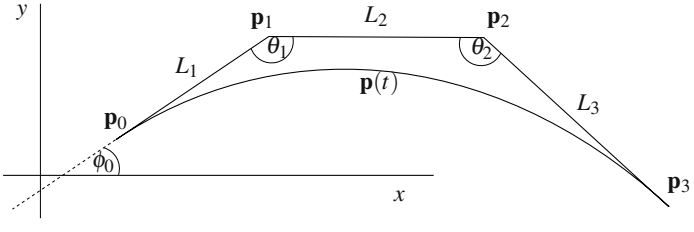


Fig. 1. Planar cubic and its control polygon

be the control polygon angles included at the interior vertices $\mathbf{p}_1$, $\mathbf{p}_2$, respectively - see Fig. 1. Then, as shown in [10], Section 4, the following theorem holds.

Theorem 9 (Farouki 1990). *Let $\mathbf{p}(t)$ be a cubic in $\mathbb{R}^2$. Then $\mathbf{p}(t)$ is a planar PH cubic if and only if*

$$L_2^2 = L_1 L_3 \quad \textit{and} \quad \theta_1 = \theta_2. \tag{8}$$

Using the PH representation map (see [6]) and the complex numbers setting we reformulate this theorem and provide an alternative proof. We express the hodograph $\mathbf{h}(t)$ of $\mathbf{p}(t)$ and its preimage $\mathscr{A}(t)$ in the Bernstein-Bézier representation

$$\mathbf{h}(t) = \sum_{i=0}^{2} \mathbf{h}_i B_i^2(t), \quad \mathscr{A}(t) = \sum_{i=0}^{1} \mathscr{A}_i B_i^1(t), \quad t \in [0,1],$$

where $\mathbf{h}_i$ and $\mathscr{A}_i$ are the complex control points.

Theorem 10. *Let $\mathbf{p}(t)$ be a cubic in $\mathbb{R}^2$. Then $\mathbf{p}(t)$ is a planar PH cubic if and only if*

$$\mathbf{h}_0 \mathbf{h}_2 = \mathbf{h}_1^2. \tag{9}$$

Proof. The relation between the hodograph and the preimage

$$\mathbf{h}(t) = \mathscr{A}(t)^2$$

can be expressed using the control points as

$$\begin{aligned} \mathbf{h}_0 &= \mathscr{A}_0^2, \\ \mathbf{h}_1 &= \mathscr{A}_0 \mathscr{A}_1, \\ \mathbf{h}_2 &= \mathscr{A}_1^2. \end{aligned} \tag{10}$$

From the basic properties of complex numbers it follows that

$$\mathbf{h}_0 \mathbf{h}_2 = \mathbf{h}_1^2. \tag{11}$$

Now for the converse statement. Let $\mathbf{p}(t)$ be a planar cubic with its hodograph control points $\mathbf{h}_i$, $i = 0,1,2$ such that it satisfies (11). First, compute $\mathscr{A}_0$ from the first equation in (10) using any complex square root. Then, compute $\mathscr{A}_1$ from $\mathbf{h}_1 = \mathscr{A}_0 \mathscr{A}_1$. Finally, one can see that the last equation in (10) holds. Therefore, $\mathscr{A}_0$ and $\mathscr{A}_1$ form preimage control points for $\mathbf{p}(t)$. Hence, $\mathbf{p}(t)$ is a planar PH cubic. □

Remark 11. Let ϕ_i be the argument of $\mathbf{h}_i$, $i = 0,1,2$, in its polar form. Then (11) is equivalent to the system

$$\begin{aligned} |\mathbf{h}_0||\mathbf{h}_2| &= |\mathbf{h}_1|^2, \\ \phi_0 + \phi_2 &= 2\phi_1, \end{aligned}$$

where $|\mathbf{h}_i|$ denotes the absolute value of the complex number $\mathbf{h}_i$. Moreover, $L_i = \frac{1}{3}|\mathbf{h}_{i-1}|$ for $i = 1,2,3$ and θ_1, θ_2 are the control polygon angles included at the interior vertices $\mathbf{p}_1$, $\mathbf{p}_2$, respectively. Thus one can see that (9) is equivalent to the system (8).

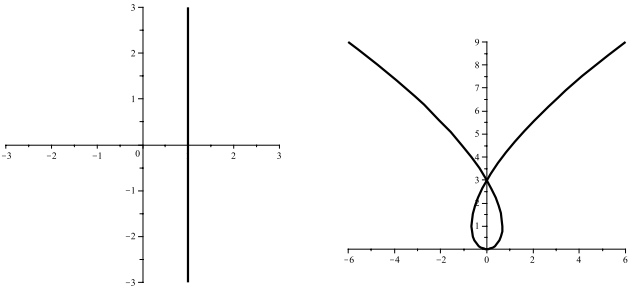


Fig. 2. The "unique" Euclidean PH cubic - the Tschirnhausen cubic (right) and its preimage (left)

Now, using the tools of Section 3, we give a new proof of the classification result for Euclidean PH cubics - see [10] for the original proof.

Theorem 12. *Up to Euclidean similarities there is only one Euclidean PH planar cubic, see Fig. 2. More precisely, any planar Euclidean PH cubic can be transformed by some element of* $CO(\mathbb{R}^2)$ *and by a reparameterization to the Tschirnhausen cubic*

$$\left(t - \frac{t^3}{3}, t^2\right)^\top. \tag{12}$$

Proof. It follows from Theorem 7 that Euclidean similarities of PH cubics can be generated by Euclidean similarities of their preimages. The preimage of any PH cubic $\mathbf{p}$ is some straight line l. We can assume that l does not pass through the origin, since if it does, the resulting curve is a straight line. By a suitable conformal transformation λ (see Definition 4) it can be mapped onto the line passing through the point $(1,0)$ and parallel to the y axis and then reparameterized so that it becomes $(1,t)^\top$, which is the preimage of the cubic (12). It follows from Theorem 7 and Lemma 6 that λ and the reparameterization of the preimage will produce a similarity which together with the reparameterization transform $\mathbf{p}$ to the desired form (12). □

4.2 MPH Cubics

Now, we move on to planar MPH cubics. Using the PH representation map (see [6]) and the Clifford algebra $\mathscr{C}\ell(1,1)$ setting we express planar MPH cubics $\mathbf{p}(t)$, their hodographs $\mathbf{h}(t)$ and the preimages $\mathscr{A}(t)$ as

$$\mathbf{p}(t) = \sum_{i=0}^{3} \mathbf{p}_i B_i^3(t), \quad \mathbf{h}(t) = \sum_{i=0}^{2} \mathbf{h}_i B_i^2(t), \quad \mathscr{A}(t) = \sum_{i=0}^{1} \mathscr{A}_i B_i^1(t), \quad t \in [0,1],$$

where $\mathbf{p}_i$, $\mathbf{h}_i$ and $\mathscr{A}_i$ are the control points. In this setting, $\mathbf{p}(t)$ and its hodograph are described by pure vectors, whereas the preimage lies in the subalgebra $\mathbb{H}^{1,1}$. The following Theorem is an analogy to the Euclidean case.

Theorem 13. *A planar cubic curve* $\mathbf{p}(t)$ *in* $\mathbb{R}^{1,1}$ *is an MPH curve if and only if its hodograph satisfies*

$$\mathbf{h}_0 \star \mathbf{h}_2 = \mathbf{h}_1 \star \mathbf{h}_1. \tag{13}$$

Proof. The relation between the hodograph and the preimage

$$\mathbf{h}(t) = \mathscr{A}(t) \star \mathscr{A}(t)$$

can be expressed using the control points as

$$\begin{aligned} \mathbf{h}_0 &= \mathscr{A}_0 \star \mathscr{A}_0, \\ \mathbf{h}_1 &= \mathscr{A}_0 \star \mathscr{A}_1, \\ \mathbf{h}_2 &= \mathscr{A}_1 \star \mathscr{A}_1. \end{aligned} \tag{14}$$

Exploiting the commutativity and associativity of the symmetric product $\star$ on pure vectors, one can see that (14) yields (13).

In order to prove the converse statement, we need to distinguish two cases, since light-like elements in $\mathscr{C}\ell(1,1)$ have no inverse elements. Let $\mathbf{p}(t)$ be a planar cubic with its hodograph control points $\mathbf{h}_i$, $i = 0,1,2$ such that it satisfies (13).

First, assume that both $\mathbf{h}_0$ and $\mathbf{h}_2$ are light-like. Then, from (13) and Lemma 2 it follows that $\mathbf{h}_1$ is light-like as well. Therefore, if $\mathscr{A}_0$ and $\mathscr{A}_1$ exist, they must be light-like too and can be written as

$$\mathscr{A}_0 = [a_0, 0, 0, \pm a_0], \quad \mathscr{A}_1 = [a_1, 0, 0, \pm a_1].$$

Checking different cases with respect to the 4 sign combinations, one can see that this preimage either gives a light-like straight line as the corresponding MPH curve, or the preimage gives $\mathbf{h}_1 = [0,0,0,0]$. In the latter case one can compute the preimage from the first and third equation of (14) using Lemma 3, thus proving that $\mathbf{p}(t)$ is an MPH cubic.

Second, let $\mathbf{h}_0$ be space-like (if $\mathbf{h}_0$ is light-like we simple interchange the roles of $\mathbf{h}_0$ and $\mathbf{h}_2$). Then, as in the Euclidean case, we compute $\mathscr{A}_0$ from the first equation of (14) using any solution, see Lemma 3. Since $N(\mathbf{h}_0) = N(\mathscr{A}_0 \star \mathscr{A}_0) = -N(\mathscr{A}_0)^2$, we have that $N(\mathscr{A}_0) \neq 0$ and $\mathscr{A}_0$ is invertible. Therefore, using the second equation of (14), we can compute

$$\mathscr{A}_1 = \frac{\mathbf{h}_1 \cdot \mathscr{A}_0 \cdot \mathbf{e}_1}{N(\mathscr{A}_0)}. \tag{15}$$

Indeed,

$$\begin{aligned} &\tfrac{\mathbf{h}_1 \cdot \mathscr{A}_0 \cdot \mathbf{e}_1}{N(\mathscr{A}_0)} \star \mathscr{A}_0 = \tfrac{1}{2N(\mathscr{A}_0)} \left(\mathbf{h}_1 \cdot \mathscr{A}_0 \cdot \mathbf{e}_1 \cdot \mathbf{e}_1 \cdot \bar{\mathscr{A}}_0 + \mathscr{A}_0 \cdot \mathbf{e}_1 \cdot \bar{\mathbf{e}}_1 \cdot \bar{\mathscr{A}}_0 \cdot \bar{\mathbf{h}}_1 \cdot \right) = \\ &\tfrac{1}{2N(\mathscr{A}_0)} \left(\mathbf{h}_1 \cdot \mathscr{A}_0 \cdot \bar{\mathscr{A}}_0 - \mathscr{A}_0 \cdot \bar{\mathscr{A}}_0 \cdot \bar{\mathbf{h}}_1 \right) = \tfrac{1}{2}(\mathbf{h}_1 - \bar{\mathbf{h}}_1) = \mathbf{h}_1. \end{aligned}$$

Now, we express

$$\begin{aligned} &\mathscr{A}_1 \star \mathscr{A}_1 = \tfrac{1}{N(\mathscr{A}_0)^2} (\mathbf{h}_1 \cdot \mathscr{A}_0 \cdot \mathbf{e}_1 \cdot \mathbf{e}_1 \cdot \bar{\mathbf{e}}_1 \cdot \bar{\mathscr{A}}_0 \cdot \bar{\mathbf{h}}_1) = \\ &-\tfrac{1}{N(\mathscr{A}_0)^2} (\mathbf{h}_1 \cdot \mathscr{A}_0 \cdot \mathbf{e}_1 \cdot \bar{\mathscr{A}}_0 \cdot \bar{\mathbf{h}}_1) = -\tfrac{1}{N(\mathscr{A}_0)^2} (\mathbf{h}_1 \cdot (\mathscr{A}_0 \star \mathscr{A}_0) \cdot \bar{\mathbf{h}}_1) = \\ &\tfrac{1}{N(h_0)} (\mathbf{h}_1 \cdot \mathbf{h}_0 \cdot \bar{\mathbf{h}}_1) \end{aligned}$$

and verify that

$$\mathbf{h}_0 \star (\mathscr{A}_1 \star \mathscr{A}_1) = \mathbf{h}_0 \star \tfrac{\mathbf{h}_1 \cdot \mathbf{h}_0 \cdot \bar{\mathbf{h}}_1}{N(\mathbf{h}_0)} = \tfrac{(\mathbf{h}_0 \cdot \mathbf{e}_1 \cdot \mathbf{h}_1 \cdot \bar{\mathbf{h}}_0 \cdot \bar{\mathbf{h}}_1 + \mathbf{h}_1 \cdot \mathbf{h}_0 \cdot \bar{\mathbf{h}}_1 \cdot \mathbf{e}_1 \cdot \bar{\mathbf{h}}_0)}{2N(\mathbf{h}_0)} = \tfrac{1}{2}(\mathbf{h}_1 \cdot \mathbf{e}_1 \cdot \bar{\mathbf{h}}_1 + \mathbf{h}_1 \cdot \mathbf{e}_1 \cdot \bar{\mathbf{h}}_1) = \mathbf{h}_1 \star \mathbf{h}_1.$$

We assumed that

$$\mathbf{h}_0 \star \mathbf{h}_2 = \mathbf{h}_1 \star \mathbf{h}_1,$$

therefore $\mathscr{A}_1 \star \mathscr{A}_1 = \mathbf{h}_2$ as $\mathbf{h}_0$ is invertible. This completes the proof, since $\mathscr{A}_0$ and $\mathscr{A}_1$ give a preimage of $\mathbf{p}(t)$. □

Still, we would like to obtain a geometric description of planar MPH curves as in the Euclidean case, see Theorem 9. However, as the Minkowski inner product is not positive definite, there is no straightforward definition of angle in the whole Minkowski plane. Nevertheless, the following Theorem holds.

Theorem 14. *Let* $\mathbf{p}(t)$ *in* $\mathbb{R}^{1,1}$ *be a planar MPH cubic. Then*

$$||\mathbf{h}_0||||\mathbf{h}_2|| = ||\mathbf{h}_1||^2 \tag{16}$$

and

$$||\mathbf{h}_2||^2 \langle \mathbf{h}_0, \mathbf{h}_1 \rangle^2 = ||\mathbf{h}_0||^2 \langle \mathbf{h}_1, \mathbf{h}_2 \rangle^2. \tag{17}$$

Proof. Comparing Clifford norms of both sides of equation (13) gives

$$N(\mathbf{h}_0 \star \mathbf{h}_2) = N(\mathbf{h}_1 \star \mathbf{h}_1).$$

As $\mathbf{h}_0$, $\mathbf{h}_1$ and $\mathbf{h}_2$ are pure vectors, applying Lemma 2 yields

$$N(\mathbf{h}_0) N(\mathbf{h}_2) = N(\mathbf{h}_1)^2.$$

As no leg of the control polygon of an MPH curve can be time-like, we arrive at (16).

Now, let $\mathscr{A}_0 = [a_0, 0, 0, a_1]$ and $\mathscr{A}_1 = [a_2, 0, 0, a_3]$ be the preimage control points of $\mathbf{p}(t)$. Using (14), one can verify that both sides of equation (17) are equal to

$$-(a_1 a_3 - a_2 a_0)^2 N(\mathscr{A}_0)^2 N(\mathscr{A}_1)^2.$$ □

Remark 15. Let L_1, L_2 and L_3 be the lengths of the control polygon legs of $\mathbf{p}(t)$ computed with respect to the Minkowski inner product, i.e. $L_i = \frac{1}{3}||\mathbf{h}_i||$. We obtained an analogous condition for the lengths of the control polygon legs of an MPH curve as in the Euclidean case (cf. (8))

$$L_2^2 = L_1 L_3.$$

In the Euclidean case, the two angles included between h_0 and h_1, h_1 and h_2 are the same, see Theorem 9. Using Euclidean inner product and norm, denoted by $_E$, one can express this condition as

$$||\mathbf{h}_2||_E \langle \mathbf{h}_0, \mathbf{h}_1 \rangle_E = ||\mathbf{h}_0||_E \langle \mathbf{h}_1, \mathbf{h}_2 \rangle_E. \tag{18}$$

Consequently, equation (17) represents an analogy to the angular formula in Theorem 9.

These observations lead us to the following definitions.

Definition 16. *Let* $\mathbf{h} = [0, h_1, h_2, 0]$ *be a pure space-like vector. If* $h_1 > 0$ *then* $\mathbf{h}$ *is called a positive space-like vector.*

Using basic geometry, one can see that any positive space-like vector can be represented using its Minkowski norm $||\mathbf{h}||$ and hyperbolic angle as

$$\mathbf{h} = [0, ||\mathbf{h}|| \cosh\phi, ||\mathbf{h}|| \sinh\phi, 0].$$

Moreover, any positive space-like vector can be uniquely represented using its Minkowski norm and hyperbolic angle.

Definition 17. *Let* $\mathbf{a}$ *and* $\mathbf{b}$ *be two positive space-like vectors. Then we define the angle* θ *included between* $\mathbf{a}$ *and* $\mathbf{b}$ *as*

$$\cosh\theta = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{||\mathbf{a}||||\mathbf{b}||}.$$

Note that this definition of angle is invariant with respect to special Lorentz transforms. Moreover, the hyperbolic angle of $\mathbf{a}$ can be defined as the angle included between $\mathbf{a}$ and $\mathbf{e}_1$.

Lemma 18. *Let* $\mathbf{a}$ *and* $\mathbf{b}$ *be two positive space-like vectors*

$$\mathbf{a} = [0, ||\mathbf{a}|| \cosh\phi, ||\mathbf{a}|| \sinh\phi, 0], \quad \mathbf{b} = [0, ||\mathbf{b}|| \cosh\psi, ||\mathbf{b}|| \sinh\psi, 0].$$

Then

$$\mathbf{a} \star \mathbf{b} = -[0, ||\mathbf{a}||||\mathbf{b}|| \cosh(\phi + \psi), ||\mathbf{a}||||\mathbf{b}|| \sinh(\phi + \psi), 0].$$

Proof. The claim is a direct consequence of the definition of the symmetric product $\star$ and the hyperbolic functions formulas

$$\begin{aligned} \cosh(\phi + \psi) &= \cosh\phi \cosh\psi + \sinh\phi \sinh\psi, \\ \sinh(\phi + \psi) &= \sinh\phi \cosh\psi + \cosh\phi \sinh\psi. \end{aligned}$$

□

Now, let us assume that $\mathbf{h}_0$, $\mathbf{h}_1$ and $\mathbf{h}_2$ are positive space-like vectors with hyperbolic angles ϕ_0, ϕ_1 and ϕ_2, respectively. Then Lemma 18 yields that both $-\mathbf{h}_0 \star \mathbf{h}_2$ and $-\mathbf{h}_1 \star \mathbf{h}_1$ are positive space-like vectors as well. Thus, we see that the equation

$$\mathbf{h}_0 \star \mathbf{h}_2 = \mathbf{h}_1 \star \mathbf{h}_1$$

is equivalent to the system

$$\begin{aligned} L_2^2 &= L_1 L_3, \\ \phi_0 + \phi_2 &= 2\phi_1. \end{aligned}$$

Finally, let θ_1, θ_2 be the angles included between $\mathbf{h}_0$ and $\mathbf{h}_1$, $\mathbf{h}_1$ and $\mathbf{h}_2$, respectively. Collecting our results, we arrive at

Theorem 19. *Let* $\mathbf{p}(t)$ *be a cubic in* $\mathbb{R}^{1,1}$ *such that its hodograph control points are positive space-like vectors. Then* $\mathbf{p}(t)$ *is a planar MPH cubic if and only if*

$$L_2^2 = L_1 L_3 \quad \textit{and} \quad \theta_1 = \theta_2. \tag{19}$$

Proof. It only remains to show that $\theta_1 = \theta_2$ is equivalent to $\phi_0 + \phi_2 = 2\phi_1$. We leave this computation as an exercise on inverse hyperbolic functions using the formulas

$$\begin{aligned} &\operatorname{arccosh}(x) = \ln(x + \sqrt{x^2-1}), \\ &\operatorname{arccosh}(x) + \operatorname{arccosh}(y) = \operatorname{arccosh}(xy + \sqrt{(x^2-1)(y^2-1)}). \end{aligned}$$

□

Now, we will give a classification of Minkowski PH planar cubics.

Theorem 20. *Up to Minkowski similarities there are only two Minkowski PH planar cubics. More precisely, any planar Minkowski PH cubic can be mapped by some element of* $CO(\mathbb{R}^{1,1})$ *on one of the following cubics - see Fig. 3*

$$\left(\frac{2}{3}t^3 + 2t, \frac{2}{3}t^3 - 2t\right)^\top, \tag{20}$$

$$\left(t + \frac{t^3}{3}, t^2\right)^\top. \tag{21}$$

Proof. Analogously to the proof of Theorem 12, we make use of Lemmas 8 and 6. The preimage of any MPH cubic $\mathbf{p}$ is a straight line l, which has one or two intersections with the light cone. If l contains only one intersection, than it is of the form (a,a) or $(a,-a)$ and we apply the mapping $\begin{pmatrix} \frac{1}{a} & 0 \\ 0 & -\frac{1}{a} \end{pmatrix}$ or $\begin{pmatrix} \frac{1}{a} & 0 \\ 0 & \frac{1}{a} \end{pmatrix}$, respectively, to map this point to the point $(1,-1)$ and the line l to the line given by $(1+t,-1+t)^\top$, which gives the

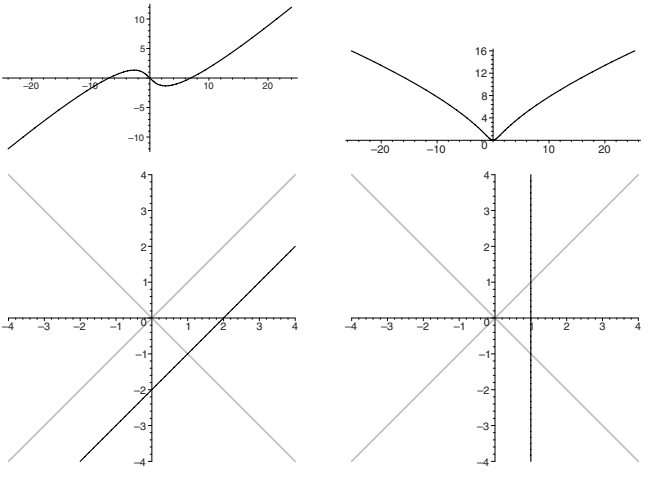


Fig. 3. Two Minkowski PH cubics (top) and their preimages (bottom)

cubic ([20]). If the straight line l has two intersections with the light cone, they are of the form (a,a) and $(b,-b)$. The mapping

$$\begin{pmatrix} (\frac{1}{2a}+\frac{1}{2b}) & (\frac{1}{2a}-\frac{1}{2b}) \\ (\frac{1}{2a}-\frac{1}{2b}) & (\frac{1}{2a}+\frac{1}{2b}) \end{pmatrix}$$

maps the line l on the line given by $(1,t)^\top$, which gives the cubic ([21]).

5 Planar PH and MPH Quintics

5.1 PH Quintics

Let $\mathbf{p}(t)$ be a planar quintic in $\mathbb{R}^2$ expressed in the Bernstein-Bézier representation [19]

$$\mathbf{p}(t) = \sum_{i=0}^{5} \mathbf{p}_i B_i^5(t), \quad t \in [0,1],$$

where $\mathbf{p}_i$, are the control points. Moreover, let L_i, $i = 1,\ldots,5$ be the lengths of the control polygon legs and θ_i, $i = 1,\ldots,4$ be the control polygon angles included at the interior vertices $\mathbf{p}_i$. Then, as observed in [10], Section 5, the following constraint on the lengths of the control polygon legs holds for any planar PH quintic

$$L_1 L_4^2 = L_5 L_2^2.$$

Using the PH representation map (see [6]) and the complex numbers setting simplifies the matter. We express the hodograph $\mathbf{h}(t)$ of $\mathbf{p}(t)$ and its preimage $\mathscr{A}(t)$ in the Bernstein-Bézier representation

$$\mathbf{h}(t) = \sum_{i=0}^{4} \mathbf{h}_i B_i^4(t), \quad \mathscr{A}(t) = \sum_{i=0}^{2} \mathscr{A}_i B_i^2(t), \quad t \in [0,1],$$

where $\mathbf{h}_i$ and $\mathscr{A}_i$ are the complex control points.

The relation between the hodograph and the preimage

$$\mathbf{h}(t) = \mathscr{A}(t)^2$$

can be expressed using the control points as

$$\begin{aligned} \mathbf{h}_0 &= \mathscr{A}_0^2, \\ \mathbf{h}_1 &= \mathscr{A}_0 \mathscr{A}_1, \\ \mathbf{h}_2 &= \tfrac{2}{3}\mathscr{A}_1^2 + \tfrac{1}{3}\mathscr{A}_0 \mathscr{A}_2, \\ \mathbf{h}_3 &= \mathscr{A}_1 \mathscr{A}_2, \\ \mathbf{h}_4 &= \mathscr{A}_2^2. \end{aligned} \tag{22}$$

From the basic properties of complex numbers it follows that

$$\mathbf{h}_0 \mathbf{h}_3^2 = \mathbf{h}_1^2 \mathbf{h}_4, \tag{23}$$

$$3\mathbf{h}_2 \mathbf{h}_3 \mathbf{h}_4 = 2\mathbf{h}_3^3 + \mathbf{h}_1 \mathbf{h}_4^2 \tag{24}$$

and

$$3\mathbf{h}_2\mathbf{h}_1\mathbf{h}_0 = 2\mathbf{h}_1^3 + \mathbf{h}_3\mathbf{h}_0^2. \tag{25}$$

Note the symmetry of the last two equations. Moreover, a simple computation reveals that (25) can be obtained from (23) and (24). More precisely, expressing e.g. $\mathbf{h}_0$ from (23) and substituting it into (24) yields (25).

Now, let ϕ_i, $i = 0,\ldots,4$ be the argument of $\mathbf{h}_i$ in its polar form and $|\mathbf{h}_i|$ its usual complex norm. Then (23) is equivalent to the system

$$\begin{aligned} |\mathbf{h}_0||\mathbf{h}_3|^2 &= |\mathbf{h}_4||\mathbf{h}_1|^2, \\ \phi_0 + 2\phi_3 &= \phi_4 + 2\phi_1. \end{aligned}$$

In turn, this system is equivalent to

$$\begin{aligned} L_1 L_4^2 &= L_5 L_2^2, \\ \theta_2 + \theta_3 &= \theta_1 + \theta_4, \end{aligned}$$

which yields a geometric constraint on the control polygon of a planar PH quintic. However, it seems that (24) (or (25)) does not admit a straightforward geometric interpretation.

Theorem 21. *Let $\mathbf{p}(t)$ be a quintic in $\mathbb{R}^2$. Then $\mathbf{p}(t)$ is a PH quintic if and only if*

$$\begin{aligned} \mathbf{h}_0\mathbf{h}_3^2 &= \mathbf{h}_1^2\mathbf{h}_4, \\ 3\mathbf{h}_2\mathbf{h}_3\mathbf{h}_4 &= 2\mathbf{h}_3^3 + \mathbf{h}_1\mathbf{h}_4^2. \end{aligned} \tag{26}$$

Proof. It only remains to show the sufficiency of the system (26). Let $\mathbf{p}(t)$ be a planar quintic with its hodograph control points $\mathbf{h}_i$, $i = 0,\ldots,4$ such that it satisfies (26). First, compute $\mathscr{A}_0$ from the first equation in (10) using any complex square root. Then, compute $\mathscr{A}_1$ from $\mathbf{h}_1 = \mathscr{A}_0\mathscr{A}_1$ and $\mathscr{A}_2$ from $\mathbf{h}_3 = \mathscr{A}_1\mathscr{A}_2$. Thus we have

$$\mathscr{A}_0^2 = \mathbf{h}_0, \quad \mathscr{A}_1 = \frac{\mathbf{h}_1}{\mathscr{A}_0}, \quad \mathscr{A}_2 = \frac{\mathbf{h}_3}{\mathscr{A}_1}.$$

Then, using the first equation of (26), we verify that

$$\mathscr{A}_2^2 = \frac{\mathbf{h}_3^2}{\mathscr{A}_1^2} = \frac{\mathbf{h}_3^2\mathbf{h}_0}{\mathbf{h}_1^2} = \mathbf{h}_4$$

and using the second equation of (26) we see that

$$\frac{2}{3}\mathscr{A}_1^2 + \frac{1}{3}\mathscr{A}_0\mathscr{A}_2 = \frac{1}{3}\left(2\frac{\mathbf{h}_1^2}{\mathbf{h}_0} + \frac{\mathbf{h}_0\mathbf{h}_3}{\mathbf{h}_1}\right) = \mathbf{h}_2.$$

Therefore, we found the control points of the preimage of $\mathbf{p}(t)$. Consequently, $\mathbf{p}(t)$ is a planar PH quintic. □

Now, we will provide a classification of Euclidean PH quintics, which is based on the methods developed in Section 3.

Theorem 22. *Any planar Euclidean PH quintic can be transformed by some element of $CO(\mathbb{R}^2)$ and a reparameterization to one of the following forms*

$$\left(-\tfrac{1}{5}t^5+\left(\tfrac{1}{3}-\tfrac{2}{3}b\right)t^3+t^2a+\left(a^2-b^2\right)t,\tfrac{1}{2}t^4+t^2b+\tfrac{2}{3}t^3a+2abt\right)^\top, \tag{27}$$

$$\left(-\tfrac{1}{5}t^5-\tfrac{2}{3}t^3c+\left(1-c^2\right)t,\tfrac{2}{3}t^3+2ct\right)^\top, \tag{28}$$

where $a\geq 0, b, c$ are real parameters.

Proof. Let $\mathbf{p}$ be a quintic PH curve and $\mathscr{A}(t)=(u(t),v(t))^\top$ its quadratic preimage. From the geometrical point of view there are two possibilities: $\mathscr{A}(t)$ is either a parabola, or it is degenerated into a half-line. In the former case we can rotate the parabola so that its axis has the direction given by $(0,1)^\top$. If its vertex x-coordinate is negative we apply the mapping $\begin{pmatrix}-1&0\\0&1\end{pmatrix}$ to make it positive. Then, we scale the parabola and reparameterize it so that the vertex corresponds to $t=0$ and it takes the form $(t+a,t^2+b)$. This preimage provides the PH curve (27). It follows from Theorem 7 and Lemma 6 that all the used transformations of the preimage will induce a similarity (element of $CO(\mathbb{R}^2)$ transforming $\mathbf{p}(t)$ to the desired form.

If $\mathscr{A}$ degenerates into a half-line, it is of the form $(a'+kq(t),b'+lq(t))^\top$, where $q(t)$ is a quadratic polynomial and a', b', l, k some real numbers. Again, we rotate $\mathscr{A}$ so that it points in the direction of $(0,1)^\top$ and possibly apply $\begin{pmatrix}-1&0\\0&1\end{pmatrix}$ in order to make the x-coordinate of its extremal point positive (it can not be 0, since this would lead to a degeneration of the quintic $\mathbf{p}$ into a line). Then, we scale it so that the x-coordinate of this point is equal to 1. Eventually, via a suitable reparameterization this half-line will take the form $(1,t^2+c)^\top$, which provides the PH curve (28). □

The explicit forms of PH quintics (27) and (28) seem to be promising for the analysis of possible singularities and other important features of PH quintics. In this paper we will restrict ourselves to a description of symmetric PH curves, which can be obtained from (28) and (27) by setting $a=0$.

Theorem 23. *Any PH quintic symmetric with respect to an axis is similar to*

$$\left(-\frac{1}{5}t^5+\left(\frac{1}{3}-\frac{2}{3}b\right)t^3-b^2t,\frac{1}{2}t^4+t^2b\right)^\top \tag{29}$$

for some $b\in\mathbb{R}$ and any centrally symmetric PH quintic is similar to

$$\left(-\frac{1}{5}t^5-\frac{2}{3}t^3c+\left(1-c^2\right)t,\frac{2}{3}t^3+2ct\right)^\top \tag{30}$$

for some $c\in\mathbb{R}$.

Proof. Obviously, (29) is symmetric with respect to the y axis and (30) with respect to the origin. Due to Theorem 22 we only need to show that for $a\neq 0$ the curve (27) is not symmetric. However, this follows from the non-symmetric position of the point on the parabola which is closest to the origin. Indeed, for its image on the PH curve there can not be any symmetric point. □

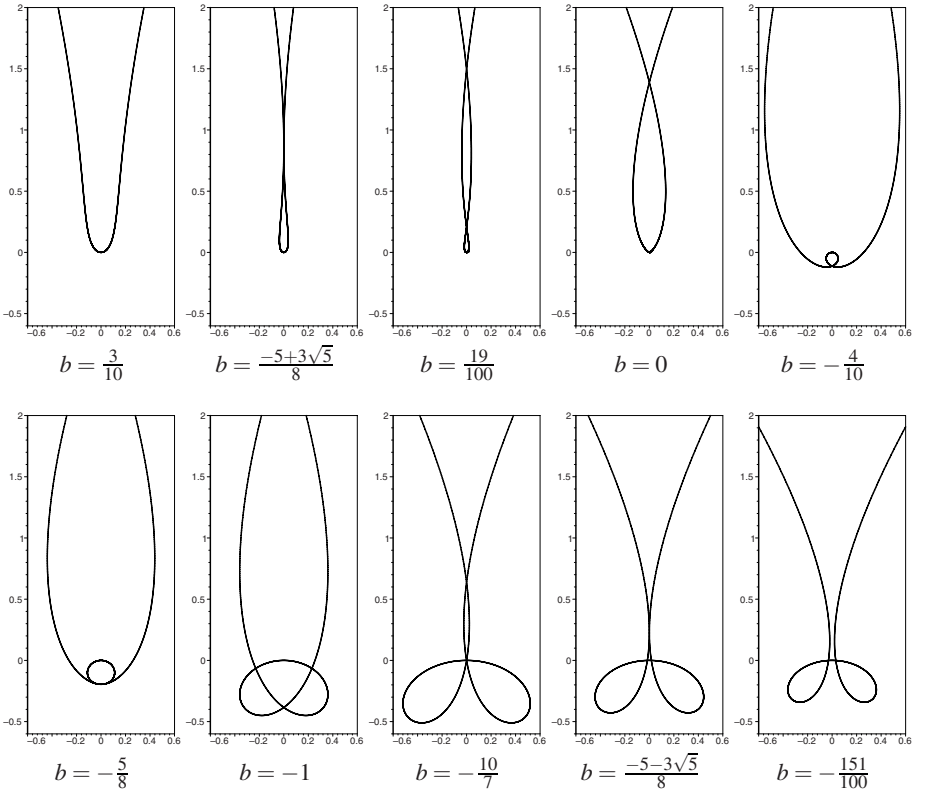


Fig. 4. Examples of PH quintics symmetric with respect to an axis

Thus, we see that symmetric PH quintics form two one-parameter systems. Figures 4 and 5 show representatives of these systems for some typical and critical values of the parameters.

5.2 MPH Quintics

Now, we move on to planar MPH quintics. Using the PH representation map and the Clifford algebra $\mathcal{C}\ell(1,1)$ setting we express planar MPH quintics $\mathbf{p}(t)$, their hodographs $\mathbf{h}(t)$ and the preimages $\mathcal{A}(t)$ as

$$\mathbf{p}(t) = \sum_{i=0}^{5} \mathbf{p}_i B_i^5(t), \quad \mathbf{h}(t) = \sum_{i=0}^{4} \mathbf{h}_i B_i^4(t), \quad \mathcal{A}(t) = \sum_{i=0}^{2} \mathcal{A}_i B_i^2(t), \quad t \in [0,1],$$

where $\mathbf{p}_i$, $\mathbf{h}_i$ and $\mathcal{A}_i$ are the control points. In this setting, $\mathbf{p}(t)$ and its hodograph are described by pure vectors, whereas the preimage lies in the subalgebra $\mathbb{H}^{1,1}$. The relation between the hodograph and the preimage

$$\mathbf{h}(t) = \mathcal{A}(t) \star \mathcal{A}(t)$$

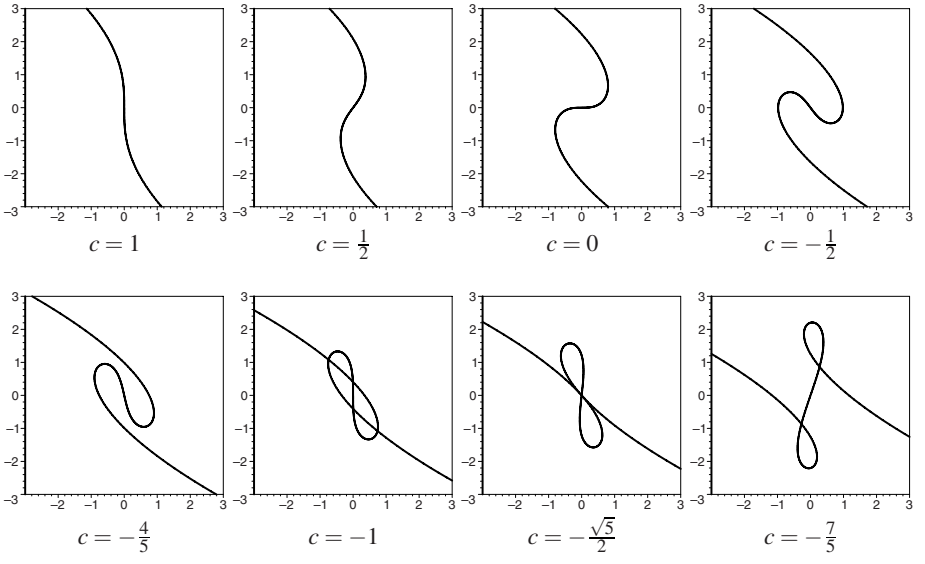


Fig. 5. Examples of PH quintics symmetric with respect to a point

can be expressed using the control points as

$$\begin{aligned}
\mathbf{h}_0 &= \mathscr{A}_0^{2\star},\\
\mathbf{h}_1 &= \mathscr{A}_0 \star \mathscr{A}_1,\\
\mathbf{h}_2 &= \tfrac{2}{3}\mathscr{A}_1^{2\star} + \tfrac{1}{3}\mathscr{A}_0 \star \mathscr{A}_2,\\
\mathbf{h}_3 &= \mathscr{A}_1 \star \mathscr{A}_2,\\
\mathbf{h}_4 &= \mathscr{A}_2^{2\star}.
\end{aligned} \tag{31}$$

Exploiting the properties of the symmetric product $\star$ as in the cubic case, one can see that

$$\mathbf{h}_0 \star \mathbf{h}_3^{2\star} = \mathbf{h}_1^{2\star} \star \mathbf{h}_4, \tag{32}$$

and

$$3\mathbf{h}_2 \star \mathbf{h}_3 \star \mathbf{h}_4 = 2\mathbf{h}_3^{3\star} + \mathbf{h}_1 \star \mathbf{h}_4^{2\star}. \tag{33}$$

The symmetric equation holds

$$3\mathbf{h}_2 \star \mathbf{h}_1 \star \mathbf{h}_0 = 2\mathbf{h}_1^{3\star} + \mathbf{h}_3 \star \mathbf{h}_0^{2\star}, \tag{34}$$

but can be derived from (32) and (33). Moreover, using the same techniques as in Section 4.2, one can show that provided that $\mathbf{h}_i$, $i = 0,\ldots,4$ are positive space–like vectors, (32) is equivalent to the system

$$\begin{aligned}
L_1 L_4^2 &= L_5 L_2^2,\\
\theta_2 + \theta_3 &= \theta_1 + \theta_4,
\end{aligned}$$

where L_i, $i = 1,\ldots,5$ are the Minkowski lengths of the control polygon legs and θ_i, $i = 1\ldots,4$ are the (Minkowski) angles (see Definition 17) at the interior vertices of the control polygon of $\mathbf{p}(t)$.

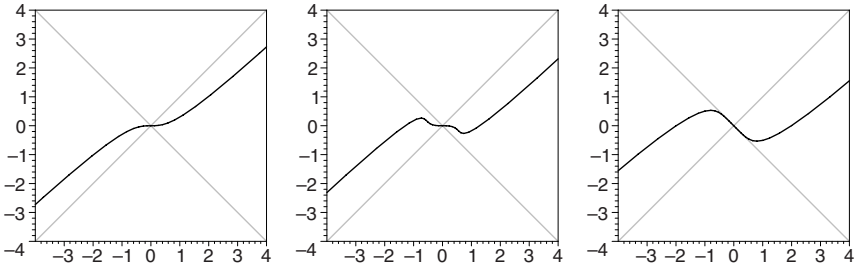


Fig. 6. Shapes of Minkowski PH quintics (42), (43) and (44)

Theorem 24. *Let* $\mathbf{p}(t)$ *be a quintic in* $\mathbb{R}^2$ *such that* $\mathbf{h}_0$ *and* $\mathbf{h}_1$ *are space–like or* $\mathbf{h}_3$ *and* $\mathbf{h}_4$ *are space–like. Then* $\mathbf{p}(t)$ *is an MPH quintic if and only if*

$$\begin{aligned} \mathbf{h}_0 \star \mathbf{h}_3^{2\star} &= \mathbf{h}_1^{2\star} \star \mathbf{h}_4, \\ 3\mathbf{h}_2 \star \mathbf{h}_3 \star \mathbf{h}_4 &= 2\mathbf{h}_3^{3\star} + \mathbf{h}_1 \star \mathbf{h}_4^{2\star}. \end{aligned} \tag{35}$$

Proof. It only remains to show the sufficiency of the system (35). Let $\mathbf{p}(t)$ be a planar quintic with its hodograph control points $\mathbf{h}_i$, $i = 0,\ldots,4$ such that it satisfies (35) and let $\mathbf{h}_0$ and $\mathbf{h}_1$ be space–like, i.e., invertible in $\mathscr{C}\ell(1,1)$. First, compute $\mathscr{A}_0$ from the first equation in (31) using Lemma 3. Then, compute $\mathscr{A}_1$ from $\mathbf{h}_1 = \mathscr{A}_0 \star \mathscr{A}_1$ and $\mathscr{A}_2$ from $\mathbf{h}_3 = \mathscr{A}_1 \star \mathscr{A}_2$. Thus we have (cf. (15))

$$\mathscr{A}_0^{2\star} = \mathbf{h}_0, \quad \mathscr{A}_1 = \frac{\mathbf{h}_1 \cdot \mathscr{A}_0 \cdot \mathbf{e}_1}{N(\mathscr{A}_0)}, \quad \mathscr{A}_2 = \frac{\mathbf{h}_3 \cdot \mathscr{A}_1 \cdot \mathbf{e}_1}{N(\mathscr{A}_1)}.$$

Now, using the following formulas

$$\mathscr{A}_1^{2\star} = \frac{\mathbf{h}_1 \cdot \mathbf{h}_0 \cdot \bar{\mathbf{h}}_1}{N(\mathbf{h}_0)}, \quad N(\mathscr{A}_1) = -\frac{N(\mathbf{h}_1)}{N(\mathscr{A}_0)}$$

we compute

$$\mathscr{A}_2^{2\star} = \frac{\mathbf{h}_3 \cdot \mathscr{A}_1 \cdot \mathbf{e}_1 \cdot \mathbf{e}_1 \cdot \bar{\mathbf{e}}_1 \cdot \bar{\mathscr{A}_1} \cdot \bar{\mathbf{h}}_3}{N(\mathscr{A}_1)^2} = -\frac{\mathbf{h}_3 \cdot \mathscr{A}_1^{2\star} \cdot \bar{\mathbf{h}}_3}{N(\mathscr{A}_1)^2} = \frac{\mathbf{h}_3 \cdot \mathbf{h}_1 \cdot \mathbf{h}_0 \cdot \bar{\mathbf{h}}_1 \cdot \bar{\mathbf{h}}_3}{N(\mathbf{h}_1)^2}.$$

Verifying that

$$\mathbf{h}_1^{2\star} \star (\mathscr{A}_2^{2\star}) = \mathbf{h}_1^{2\star} \star \mathbf{h}_4,$$

using the first equation of (35) and the invertibility of $\mathbf{h}_1$ we conclude that

$$\mathscr{A}_2^{2\star} = \mathbf{h}_4.$$

Second, we need to show that

$$\frac{2}{3}\mathscr{A}_1^{2\star} + \frac{1}{3}\mathscr{A}_0 \star \mathscr{A}_2 = \mathbf{h}_2. \tag{36}$$

In order to do that, we compute

$$\begin{aligned} \mathscr{A}_2 &= \tfrac{\mathbf{h}_3 \cdot \mathscr{A}_1 \cdot \mathbf{e}_1}{N(\mathscr{A}_1)} = \tfrac{\mathbf{h}_3 \cdot \mathbf{h}_1 \cdot \mathscr{A}_0}{N(\mathscr{A}_0) N(\mathscr{A}_1)}, \\ \mathscr{A}_0 \star \mathscr{A}_2 &= \tfrac{\mathscr{A}_0 \cdot \mathbf{e}_1 \cdot \bar{\mathscr{A}_0} \cdot \bar{\mathbf{h}}_1 \cdot \bar{\mathbf{h}}_3 + \mathbf{h}_3 \cdot \mathbf{h}_1 \cdot \mathscr{A}_0 \cdot \mathbf{e}_1 \cdot \bar{\mathscr{A}_0}}{2N(\mathscr{A}_0) N(\mathscr{A}_1)} = \tfrac{\mathbf{h}_0 \cdot \bar{\mathbf{h}}_1 \cdot \bar{\mathbf{h}}_3 + \mathbf{h}_3 \cdot \mathbf{h}_1 \cdot \mathbf{h}_0}{2N(\mathscr{A}_0) N(\mathscr{A}_1)}. \end{aligned}$$

Thus we have

$$\frac{2}{3}\mathscr{A}_1^{2\star}+\frac{1}{3}\mathscr{A}_0\star\mathscr{A}_2=\frac{2\mathbf{h}_1\cdot\mathbf{h}_0\cdot\bar{\mathbf{h}}_1}{3N(\mathbf{h}_0)}+\frac{\mathbf{h}_0\cdot\bar{\mathbf{h}}_1\cdot\bar{\mathbf{h}}_3+\mathbf{h}_3\cdot\mathbf{h}_1\cdot\mathbf{h}_0}{6N(\mathscr{A}_0)N(\mathscr{A}_1)}.$$

Finally, by a straightforward computation we arrive at

$$\left(\frac{2}{3}\mathscr{A}_1^{2\star}+\frac{1}{3}\mathscr{A}_0\star\mathscr{A}_2\right)\star\mathbf{h}_1\star\mathbf{h}_0=2\mathbf{h}_1^{3\star}+\mathbf{h}_3\star\mathbf{h}_0^{2\star}.$$

Comparing this equation with (34) gives (36) since $\mathbf{h}_0$ and $\mathbf{h}_1$ are invertible. Thus we found the control points of the preimage of $\mathbf{p}(t)$. Consequently, $\mathbf{p}(t)$ is a planar MPH quintic. If $\mathbf{h}_3$ and $\mathbf{h}_4$ are space–like, we simply use the symmetry of the equations (32), (33) and (34). □

Now, making use of the methods developed in Section 3, we will give a classification of Minkowski PH quintics.

Theorem 25. *Any planar Minkowski PH quintic can be transformed by some element of $CO(\mathbb{R}^{1,1})$ and a reparameterization to one of the following forms*

$$\left(\frac{1}{5}t^5+\left(\frac{1}{3}+\frac{2}{3}b\right)t^3+t^2a+\left(a^2+b^2\right)t,\frac{1}{2}t^4+t^2b+\frac{2}{3}t^3a+2abt\right)^\top, \tag{37}$$

$$\left(\frac{1}{5}t^5+\frac{2}{3}t^3c+\left(1+c^2\right)t,\frac{2}{3}t^3+2ct\right)^\top, \tag{38}$$

$$\left(\frac{2}{5}t^5+\left(\frac{2}{3}+\frac{4}{3}c\right)t^3+2c^2t,\frac{2}{5}t^5+\left(-\frac{2}{3}+\frac{4}{3}c\right)t^3+2c^2t\right)^\top, \tag{39}$$

$$\left(\frac{2}{5}t^5+\left(\frac{2}{3}+\frac{2}{3}c\right)t^3-ct^2+c^2t,\frac{2}{5}t^5+\left(-\frac{2}{3}+\frac{2}{3}c\right)t^3+ct^2\right)^\top, \tag{40}$$

$$\left(\frac{2}{5}t^5+\frac{2}{3}t^3-2dt^2+2d^2t,\frac{2}{5}t^5-\frac{2}{3}t^3+2dt^2-2d^2t\right)^\top, \tag{41}$$

$$\left(\frac{2}{5}t^5+\frac{2}{3}t^3+t,\frac{2}{5}t^5+\frac{2}{3}t^3-t\right)^\top, \tag{42}$$

$$\left(\frac{2}{5}t^5-\frac{2}{3}t^3+t,\frac{2}{5}t^5-\frac{2}{3}t^3-t\right)^\top, \tag{43}$$

$$\left(\frac{2}{5}t^5+2t,\frac{2}{5}t^5-2t\right)^\top, \tag{44}$$

where $a\geq 0$, b, c and $d>0$ are real parameters.

Proof. The proof is analogous to the proof of Theorem 22. Let $\mathbf{p}$ be a quintic MPH curve and $\mathscr{A}(t)=(u(t),v(t))^\top$ its quadratic preimage. If the preimage is a parabola whose axis has a non-isotropic direction, then via the action of the group $PCO(\mathbb{R}^{1,1})$ and a reparameterization it can be transformed into the form $(t+a,t^2+b)$, which gives

the quintic (37). If $\mathscr{A}$ degenerates into a half-line with a non-isotropic direction, it can be transformed into the form $(1,t^2+c)^\top$, which provides the MPH curve (38).

The remaining cases correspond to the parabola with its axis direction given by $(1,1)^\top$ or a half line with the same direction. This orientation can not be changed by any element of $PCO(\mathbb{R}^{1,1})$ to a non-isotropic one. These elements can however move the vertex (the half-line end point) of the preimage so that it lies on one of the lines $u=v$, $v=0$ or $u=-v$. After a scaling and a reparameterization we obtain the following possible preimages for the parabola: $(c-t+t^2,c+t+t^2)^\top$ giving (39), $(c-t+t^2,t+t^2)^\top$ yielding (40) and $(d-t+t^2,-d+t+t^2)^\top$ giving (41). In the case of half-line, a suitable scaling removes any remaining free parameters and we arrive at three cases of preimages: $(1+t^2,t^2)^\top$ giving (42), $(-1+t^2,t^2)^\top$, which yields (43) and $(1+t^2,-1+t^2)^\top$, which gives (44). □

As we have seen in the proof, the main classes of Minkowski PH quintic are (37), (38). The other cases have preimages with particular positions with respect to the iso-tropic lines. To conclude this section we give figures of the three particular quintics (42)–(44), which are the only centrally symmetrical Minkowski PH quintics having preimages parallel to isotropic lines - see Fig. 6.

6 Conclusion

The main contributions of this paper are of two types. First, in the case of Euclidean PH cubics we provide new simple proofs for the known facts [10] about their control polygons and their classification. For Minkowski PH curves the classification result was already published in [23]. We give a simplified proof. Moreover, we give an original sufficient and necessary condition for the control polygon. Results for quintics in both Euclidean and Minkowski planes are new; we provide classifications and sufficient and necessary conditions for control polygons of quintic PH curves.

Second, the concept of the *preimage of a transformation* was introduced in a general way and it seems to be very promising for studying PH curves of even higher degrees. In particular we plan to give an analysis of possible configurations of singularities for planar PH quintics and to classify low degree curves in the Euclidean and Minkowski three dimensional spaces.

Acknowledgment

Zbyněk Šír was supported by the project MSM 0021620839 of the Czech Ministry of Education and by the project no. 201/08/0486 of the Czech Science Foundation. We thank all referees for their comments which helped us to improve the paper.

References

1. Ahn, M.-H., Kim, G.-I., Lee, C.-N.: Geometry of root-related parameters of PH curves. Appl. Math. Lett. 16, 49–57 (2003)
2. Aigner, M., Šír, Z., Jüttler, B.: Evolution-based least-squares fitting using Pythagorean hodograph spline curves. Comput. Aided Geom. Design 24, 310–322 (2007)

3. Cho, H.C., Choi, H.I., Kwon, S.-H., Lee, D.S., Wee, N.-S.: Clifford algebra, Lorentzian geometry and rational parametrization of canal surfaces. Comp. Aided Geom. Design 21, 327–339 (2004)
4. Choi, H.I., Han, C.Y.: Euler Rodrigues frames on spatial Pythagorean-hodograph curves. Comput. Aided Geom. Design 19, 603–620 (2002)
5. Choi, H.I., Han, C.Y., Moon, H.P., Roh, K.H., Wee, N.S.: Medial axis transform and offset curves by Minkowski Pythagorean hodograph curves. Comp.–Aided Design 31, 59–72 (1999)
6. Choi, H.I., Lee, D.S., Moon, H.P.: Clifford algebra, spin representation and rational parameterization of curves and surfaces. Advances in Computational Mathematics 17, 5–48 (2002)
7. Dietz, R., Hoschek, J., Jüttler, B.: An algebraic approach to curves and surfaces on the sphere and on other quadrics. Comp. Aided Geom. Design 10, 211–229 (1993)
8. Farouki, R.T.: The conformal map $z \to z^2$ of the hodograph plane. Comp. Aided Geom. Design 11, 363–390 (1994)
9. Farouki, R.T., Neff, C.A.: Hermite interpolation by Pythagorean-hodograph quintics. Math. Comput. 64, 1589–1609 (1995)
10. Farouki, R.T., Sakkalis, T.: Pythagorean hodographs. IBM Journal of Research and Development 34, 736–752 (1990)
11. Farouki, R.T., Sakkalis, T.: Pythagorean-hodograph space curves, Adv. Comput. Math. 2, 41–66 (1994)
12. Farouki, R.T., Manjunathaiah, J., Jee, S.: Design of rational cam profiles with Pythagorean-hodograph curves. Mech. Mach. Theory 33, 669–682 (1998)
13. Farouki, R.T., Saitou, K., Tsai, Y.-F.: Least-squares tool path approximation with Pythagorean-hodograph curves for high-speed CNC machining. The Mathematics of Surfaces VIII, pp. 245–264. Information Geometers, Winchester (1998)
14. Farouki, R.T., al-Kandari, M., Sakkalis, T.: Hermite interpolation by rotation–invariant spatial Pythagorean–hodograph curves. Advances in Computational Mathematics 17, 369–383 (2002)
15. Farouki, R.T.: Pythagorean–hodograph curves. In: Hoschek, J., Farin, G., Kim, M.-S. (eds.) Handbook of Computer Aided Geometric Design, pp. 405–427. Elsevier, Amsterdam (2002)
16. Farouki, R.T., Manni, C., Sestini, A.: Spatial C^2 PH quintic splines. In: Curve and surface design, Saint-Malo, 2002. Mod. Methods Math., pp. 147–156. Nashboro Press (2003)
17. Farouki, R.: Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable. Springer, Berlin (2008)
18. Farouki, R.T., al-Kandari, M., Sakkalis, T.: Structural invariance of spatial Pythagorean hodographs. Comp. Aided Geom. Design 19, 395–407 (2002)
19. Hoschek, J., Lasser, D.: Fundamentals of Computer Aided Geometric Design. AK Peters, Wellesley (1996)
20. Jüttler, B.: Hermite interpolation by Pythagorean hodograph curves of degree seven. Math. Comp. 70, 1089–1111 (2001)
21. Jüttler, B., Mäurer, C.: Cubic Pythagorean Hodograph Spline Curves and Applications to Sweep Surface Modeling. Comp.–Aided Design 31, 73–83 (1999)
22. Kim, G.–I., Ahn, M.–H.: C^1 Hermite interpolation using MPH quartic. Comput. Aided Geom. Des. 20, 469–492 (2003)
23. Kosinka, J., Jüttler, B.: Cubic Helices in Minkowski space, Sitzungsber. Österr. Akad. Wiss., Abt. II 215, 13–35 (2006)
24. Kosinka, J., Jüttler, B.: G^1 Hermite Interpolation by Minkowski Pythagorean Hodograph Cubics. Comp. Aided Geom. Des. 23, 401–418 (2006)
25. Kosinka, J., Jüttler, B.: C^1 Hermite Interpolation by Pythagorean Hodograph Quintics in Minkowski space. Advances in Applied Mathematics 30, 123–140 (2009)

26. Kubota, K.K.: Pythagorean triples in unique factorization domains. American Mathematical Monthly 79, 503–505 (1972)
27. Lounesto, P.: Clifford Algebras and Spinors. London Mathematical Society Lecture Note Series (No. 286). Cambridge University Press, Cambridge (2001)
28. Meek, D.S., Walton, D.J.: Geometric Hermite interpolation with Tschirnhausen cubics. Journal of Computational and Applied Mathematics 81, 299–309 (1997)
29. Moon, H.P.: Minkowski Pythagorean hodographs. Computer Aided Geometric Design 16, 739–753 (1999)
30. Moon, H.P., Farouki, R.T., Choi, H.I.: Construction and shape analysis of PH quintic Hermite interpolants. Comp. Aided Geom. Design 18, 93–115 (2001)
31. Pottmann, H., Peternell, M.: Applications of Laguerre geometry in CAGD. Comp. Aided Geom. Design 15, 165–186 (1997)
32. Šír, Z., Jüttler, B.: Euclidean and Minkowski Pythagorean hodograph curves over planar cubics. Comput. Aided Geom. Des. 22(8), 753–770 (2005)
33. Šír, Z., Jüttler, B.: C^2 Hermite interpolation by spatial Pythagorean hodograph curves. Math. Comp. 76, 1373–1391 (2007)
34. Walton, D.J., Meek, D.S.: G^2 curve design with a pair of Pythagorean hodograph quintic spiral segments (English summary) Comput. Aided Geom. Design 24(5), 267–285 (2007)

On the Local Approximation Power of Quasi-Hierarchical Powell-Sabin Splines

Hendrik Speleers, Paul Dierckx, and Stefan Vandewalle

Department of Computer Science, Katholieke Universiteit Leuven
Celestijnenlaan 200A, B-3001 Leuven, Belgium

Abstract. Quasi-hierarchical Powell-Sabin (QHPS) splines are quadratic splines with a global C^1-continuity. They are defined on a locally refined hierarchical triangulation, and they admit a compact representation in a normalized B-spline basis. We show that sufficiently smooth functions and their derivatives can be approximated up to optimal order by a Hermite interpolating QHPS spline.

1 Introduction

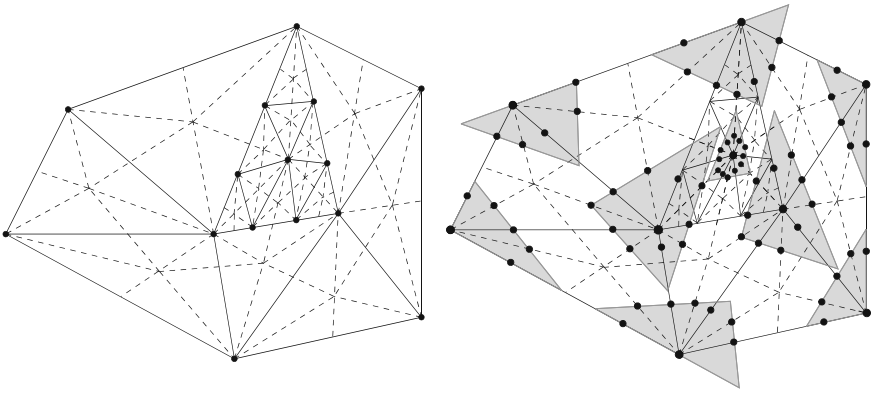
Powell-Sabin splines [1] are C^1-continuous quadratic splines defined on a conforming triangulation. They can be compactly represented in a normalized B-spline basis with an intuitive geometric interpretation involving control triangles [2]. These splines are effective in many application domains, such as surface modelling [3] and data fitting [4,5].

Recently, we developed a hierarchical variant of Powell-Sabin splines in a quasi-hierarchical representation, the so-called QHPS splines [6]. A hierarchical basis is constructed such that the basis functions of the coarser spaces are retained in the basis of the finer space. In a quasi-hierarchical basis some of the coarse-level basis functions are replaced by finer-level basis functions [7]. Similar to classical hierarchical Powell-Sabin splines, the QHPS splines admit a local refinement in a straightforward way. In addition, they have some other advantages. The basis functions form a convex partition of unity. With each B-spline one can associate a control point such that the QHPS surface lies inside the convex hull of these points. The support of the basis functions is also smaller than in case of a hierarchical basis. The basis is shown to be strongly stable on uniform hierarchical triangulations.

In this paper we consider the approximation power of QHPS splines. Splines of degree d, defined on mesh Δ, have an optimal approximation order, if there exists an approximating spline for any sufficiently smooth function, where the approximation error is of the order $|\Delta|^{d+1}$ with $|\Delta|$ the length of the longest edge in Δ. For (quasi-) hierarchical splines it is only meaningful to define a local approximation order, since the mesh can be locally refined. It means for QHPS splines that the approximation error on each triangle $\mathcal{T}$ in Δ must be of the order $|\mathcal{T}|^3$, where $|\mathcal{T}|$ is the length of the longest edge of $\mathcal{T}$.

The paper is organized as follows. Section 2 recalls the definition of the QHPS spline space, and the construction of its basis. In Sect. 3 we establish a number of useful properties of polynomials. We derive in Sect. 4 an error bound for QHPS spline approximation, and it is illustrated numerically in Sect. 5.

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 419–433, 2010.



(a) (b)

Fig. 1. (a) A HPS refinement Δ_H^* (dashed lines) of a given hierarchical triangulation Δ_H (solid lines). (b) The QHPS points (bullets) and a set of suitable QHPS triangles (shaded).

2 Quasi-Hierarchical Powell-Sabin Splines

Consider a simply connected subset $\Omega \subset \mathbb{R}^2$ with polygonal boundary, and assume a conforming triangulation Δ^0 of Ω is given. A triangulation is conforming if no triangle contains a vertex different from its own three vertices. We construct a hierarchical triangulation Δ_H on Ω by partitioning successively subsets of triangles with a triadic split, starting from the initial triangulation Δ^0. An example of such a triangulation is drawn in Fig. 1(a) with solid lines. The hierarchical triangulation has a total of n vertices. Of these vertices, n_h are non-conforming (or hanging) vertices. They are located on interiors of triangle edges. The remaining ones, i.e. $n_c = n - n_h$, are called conforming vertices.

To each hierarchical triangulation Δ_H we can associate a hierarchical structure $\mathbf{\Delta}_H$. It is the set of triangles $\mathcal{T}^k$ that are generated during the construction of Δ_H. The superscript k of a triangle $\mathcal{T}^k$ refers to the refinement level of that triangle, i.e., the minimal number of refinement steps needed to construct the triangle. We will denote $\mathbf{\Delta}_H^l$ as the subset of $\mathbf{\Delta}_H$ containing all triangles whose level number is l or lower, and let Δ_H^l be its corresponding hierarchical triangulation. Note that these hierarchical structures are nested

$$\Delta^0 \subset \mathbf{\Delta}_H^1 \subset \mathbf{\Delta}_H^2 \subset \ldots \subset \mathbf{\Delta}_H. \tag{1}$$

The hierarchical Powell-Sabin (HPS) refinement Δ_H^* is a particular partition of Δ_H, where every triangle is split into six smaller triangles with a common vertex (see [6] for details on the construction). In Fig. 1(a) such a HPS refinement is drawn in dashed lines. Analogous to (1), the HPS refinement yields a nested structure of sets of triangles

$$\Delta^{*,0} \subset \mathbf{\Delta}_H^{*,1} \subset \mathbf{\Delta}_H^{*,2} \subset \ldots \subset \mathbf{\Delta}_H^*. \tag{2}$$

Let Π_d be the space of bivariate polynomials of degree less than or equal to d. The space of piecewise quadratic polynomials on Δ_H^* with global C^1-continuity is called the hierarchical Powell-Sabin spline space:

$$S^1_{2,H}(\Delta^*_H) = \left\{ s_H \in C^1(\Omega) : s_H|_{\mathcal{T}^*_j} \in \Pi_2,\ \mathcal{T}^*_j \in \Delta^*_H \right\}. \tag{3}$$

In [6] is proved that the following interpolation problem at each conforming vertex V_k

$$s_H(V_k) = f_k,\ D_x s_H(V_k) = f_{x,k},\ D_y s_H(V_k) = f_{y,k}, \tag{4}$$

has a unique solution $s_H(x,y) \in S^1_{2,H}(\Delta^*_H)$ for any set of n_c triplets $(f_k, f_{x,k}, f_{y,k})$. Hence, the dimension of the hierarchical Powell-Sabin spline space is equal to $3n_c$.

We presented a geometric method in [6] to construct a normalized quasi-hierarchical basis for the spline space $S^1_{2,H}(\Delta^*_H)$. A hierarchical Powell-Sabin spline in its quasi-hierarchical representation is called a QHPS spline, and it is denoted as

$$s_{QH}(x,y) = \sum_{i=1}^{n_c} \sum_{j=1}^{3} c_{i,j} B^j_{i,QH}(x,y). \tag{5}$$

To find the QHPS basis functions, we associate with each conforming vertex V_i in the hierarchical triangulation three linearly independent triplets $(\alpha_{i,j}, \beta_{i,j}, \gamma_{i,j})$, $j = 1,2,3$. These triplets are determined as follows:

1. For each conforming vertex V_i in the hierarchical triangulation Δ_H, identify the corresponding QHPS points. Let k be the smallest level number of all triangles in Δ_H that contain vertex V_i. The QHPS points of V_i are defined as the midpoints of all edges in the HPS refinement $\Delta^{*,k}_H$ ending in V_i. The vertex V_i itself is also a QHPS point. In Fig. 1(b) the QHPS points are indicated as bullets.
2. For each conforming vertex V_i, find a triangle $t_i(Q_{i,1}, Q_{i,2}, Q_{i,3})$ containing all the QHPS points of V_i. The triangles t_i, $i = 1, \ldots, n_c$, are called QHPS triangles. Note that these triangles are not uniquely defined. Figure 1(b) shows a possible set of QHPS triangles.
3. The three linearly independent triplets $(\alpha_{i,j}, \beta_{i,j}, \gamma_{i,j})$, $j = 1,2,3$, are derived from the QHPS triangle t_i of a vertex V_i as follows:
   - $\alpha_i = (\alpha_{i,1}, \alpha_{i,2}, \alpha_{i,3})$ are the barycentric coordinates of V_i with respect to t_i,
   - $\beta_i = (\beta_{i,1}, \beta_{i,2}, \beta_{i,3})$ and $\gamma_i = (\gamma_{i,1}, \gamma_{i,2}, \gamma_{i,3})$ are the coordinates, with respect to t_i, of the unit barycentric directions, in the x- and y-direction respectively, with respect to t_i.

The QHPS B-spline $B^j_{i,QH}(x,y)$ is defined as the unique solution of interpolation problem (4) with all $(f_k, f_{x,k}, f_{y,k}) = (0,0,0)$ except for $k = i$, where

$$(f_i, f_{x,i}, f_{y,i}) = (\alpha_{i,j}, \beta_{i,j}, \gamma_{i,j}) \neq (0,0,0).$$

These quasi-hierarchical basis functions satisfy some interesting properties. The QHPS B-splines have a local support. We denote the support of the B-splines associated with the conforming vertex V_i as $\mathcal{S}_i$ and its area as $\mathcal{A}_{\mathcal{S}_i}$. The proposed basis also forms a convex partition of unity on Ω, i.e.,

$$B^j_{i,QH}(x,y) \geq 0, \quad \text{and} \quad \sum_{i=1}^{n_c} \sum_{j=1}^{3} B^j_{i,QH}(x,y) = 1. \tag{6}$$

For any coefficient vector $C = \{c_{i,j} : i = 1, \ldots, n_c; j = 1, 2, 3\}$ in the QHPS spline representation (5) it holds that

$$K_\infty \|C\|_\infty \leq \|s_{QH}(x, y)\|_\infty \leq \|C\|_\infty, \tag{7}$$

where constant K_∞ only depends on the smallest angle in the triangulation and on the size of the QHPS triangles. Moreover, the smaller the QHPS triangles, the larger the stability constant. A common choice of QHPS triangles are the so-called optimal triangles, i.e., the triangles with minimal area. On uniform hierarchical triangulations K_∞ is independent of the number of levels in the hierarchical structure, such that the corresponding QHPS basis is strongly stable [6]. Stability for the more general L_p-norm is considered in [8].

We find QHPS control points as $\mathbf{c}_{i,j} = (Q_{i,j}, c_{i,j})$. These points define QHPS control triangles $T_i(\mathbf{c}_{i,1}, \mathbf{c}_{i,2}, \mathbf{c}_{i,3})$, which are tangent to the spline surface $z = s_{QH}(x, y)$ at the conforming vertices V_i. The projection of the control triangles T_i in the (x, y)-plane are the QHPS triangles t_i.

If the hierarchical triangulation is obtained by global triadic splits, i.e., the final triangulation is conforming, then the QHPS B-splines are just the classical normalized Powell-Sabin B-splines [2].

3 Polynomial Approximation

We first consider some inequalities for the L_p-norm of polynomials on a triangle $\mathcal{T} \subset \mathbb{R}^2$. This norm will be denoted by the symbol $\|\cdot\|_{p,\mathcal{T}}$. Let $|\mathcal{T}|$ be the length of the longest edge of triangle $\mathcal{T}$, and let $r_\mathcal{T}$ be its inradius.

Lemma 1. *Let* $\|\nabla p_d\|_{\infty,\mathcal{T}} = \|\max\{|D_x p_d|, |D_y p_d|\}\|_{\infty,\mathcal{T}}$. *Then, for any* $p_d \in \Pi_d(\mathcal{T})$ *it holds that*

$$\|\nabla p_d\|_{\infty,\mathcal{T}} \leq \frac{2d^2 - d}{r_\mathcal{T}} \|p_d\|_{\infty,\mathcal{T}}. \tag{8}$$

Proof. From [9] we know that

$$\|\nabla p_d\|_{\infty,\mathcal{T}} \leq \frac{4d^2 - 2d}{\omega_\mathcal{T}} \|p_d\|_{\infty,\mathcal{T}},$$

with $\omega_\mathcal{T}$ the minimal distance between two parallel supporting lines that contain triangle $\mathcal{T}$ between them. Using inequality $2r_\mathcal{T} \leq \omega_\mathcal{T}$ completes the proof. □

Lemma 2. *For any* $p_d \in \Pi_d(\mathcal{T})$ *it holds that*

$$\|D_x^\alpha D_y^\beta p_d\|_{\infty,\mathcal{T}} \leq \left(\frac{2d^2 - d}{r_\mathcal{T}}\right)^{\alpha+\beta} \|p_d\|_{\infty,\mathcal{T}}, \tag{9}$$

with $0 \leq \alpha + \beta \leq d$.

Proof. The result for $\alpha + \beta = 0$ is trivial, and Lemma 1 proves the case $\alpha + \beta = 1$. The inequality for general α and β follows by applying Lemma 1 for the D_x and D_y derivatives repeatedly. □

Lemma 3. *For any $p_d \in \Pi_d(\mathcal{T})$ we have*

$$\|p_d\|_{\infty,\mathcal{T}} \leq \left(\frac{12\,(4d^2 - 2d)^2}{\mathcal{A}_{\mathcal{T}}} \right)^{1/p} \|p_d\|_{p,\mathcal{T}}, \tag{10}$$

with $\mathcal{A}_{\mathcal{T}}$ the area of $\mathcal{T}$.

Proof. For the proof we refer to [8]. □

We now introduce the so-called averaged Taylor polynomials, and we consider some of their properties. More details can be found in [10].

Let $\mathcal{B} = \{(x, y) : (x - x_0)^2 + (y - y_0)^2 < r^2\}$ be the disk centered around the point (x_0, y_0) with radius r. The function

$$g_{\mathcal{B}}(x, y) = \begin{cases} K\, e^{-\left(1-((x-x_0)^2+(y-y_0)^2)/r^2\right)^{-1}}, & \text{if } (x, y) \in \mathcal{B}, \\ 0, & \text{otherwise,} \end{cases}$$

is called a cut-off function with constant K chosen such that $\int_{\mathbb{R}^2} g_{\mathcal{B}}(x, y)\, dxdy = 1$. For any function $f \in C^d(\mathbb{R}^2)$, we recall that

$$T_{d,(u,v)} f(x, y) = \sum_{\alpha+\beta \leq d} \frac{D_u^\alpha D_v^\beta f(u, v)}{\alpha!\, \beta!} (x - u)^\alpha (y - v)^\beta \tag{11}$$

is the Taylor polynomial of degree d of f at (u, v).

Definition 4. *The averaged Taylor polynomial of degree d over disk $\mathcal{B}$ is defined as*

$$F_{d,\mathcal{B}} f(x, y) = \int_{\mathcal{B}} T_{d,(u,v)} f(x, y)\, g_{\mathcal{B}}(u, v)\, dudv \tag{12}$$

with $T_{d,(u,v)} f(x, y)$ the Taylor polynomial (11) and $g_{\mathcal{B}}(u, v)$ the cut-off function supported in the disk $\mathcal{B}$.

The averaged Taylor polynomial is well-defined for any integrable function $f \in L_1(\mathcal{B})$. It can be verified that $F_{d,\mathcal{B}} f(x, y)$ is a polynomial of degree less than or equal to d. It holds that

$$F_{d,\mathcal{B}}\, p_d = p_d, \qquad \forall p_d \in \Pi_d. \tag{13}$$

Let $|\cdot|_{k,p,\Omega}$ be the usual semi-norm in the Sobolev space $W_p^k(\Omega)$, i.e.,

$$|f|_{k,p,\Omega} = \left(\sum_{\alpha+\beta=k} \|D_x^\alpha D_y^\beta f\|_{p,\Omega}^p \right)^{1/p}. \tag{14}$$

The next lemmas concerning averaged Taylor polynomials will be needed in the proof of the approximation theorem for QHPS splines (Theorem 11). The first two lemmas are recalled from [10]. Lemma 7 is a particular modification of Proposition 4.3.2 in [10].

Lemma 5. *For any* $0 \leq \alpha + \beta \leq d$ *and* $f \in W_1^{\alpha+\beta}(\mathcal{B})$*, we have*

$$D_x^\alpha D_y^\beta F_{d,\mathcal{B}} f = F_{d-\alpha-\beta,\mathcal{B}}(D_x^\alpha D_y^\beta f). \tag{15}$$

Lemma 6. *Let* $\mathcal{B}_\mathcal{T}$ *be the largest disk in triangle* $\mathcal{T}$ *with radius* $r_\mathcal{T}$*. For any* $f \in W_p^{d+1}(\mathcal{T})$*, it holds for* $1 \leq p \leq \infty$ *and* $0 \leq \alpha + \beta \leq d$ *that*

$$\|D_x^\alpha D_y^\beta (f - F_{d,\mathcal{B}_\mathcal{T}} f)\|_{p,\mathcal{T}} \leq K_{t,p} \, |\mathcal{T}|^{d+1-\alpha-\beta} |f|_{d+1,p,\mathcal{T}}. \tag{16}$$

The constant $K_{t,p}$ *is independent of* $|\mathcal{T}|$ *and* f*.*

Lemma 7. *Let* $\mathcal{B}$ *be a disk in triangle* $\mathcal{T}$ *with centre* (x_0, y_0) *and radius* r*. For any* $f \in W_p^{d+1}(\mathcal{T})$*, it holds that*

$$\|f - F_{d,\mathcal{B}} f\|_{\infty,\mathcal{T}} \leq K_t \left(1 + \frac{|\mathcal{T}|}{r}\right)^2 |\mathcal{T}|^{d+1-2/p} |f|_{d+1,p,\mathcal{T}}, \tag{17}$$

provided that $1 < p < \infty$ *and* $d > 2/p - 1$*, or* $p = 1$ *and* $d \geq 1$*, or* $p = \infty$ *and* $d \geq 0$*. The constant* K_t *is independent of* $|\mathcal{T}|$ *and* f*.*

Proof. From [10] we know that the remainder $f - F_{d,\mathcal{B}} f$ has the following pointwise representation for any point $(x, y) \in \mathcal{T}$:

$$\begin{aligned}
&(f - F_{d,\mathcal{B}} f)(x, y) \\
&\quad = (d+1) \sum_{\alpha+\beta=d+1} \int_{\mathcal{B}_{(x,y)}} \frac{(x-u)^\alpha (y-v)^\beta}{\alpha!\beta!} \, k(x, y, u, v) \, D_u^\alpha D_v^\beta f(u, v) \, du dv,
\end{aligned}$$

where $\mathcal{B}_{(x,y)}$ is the domain defined by the convex hull of $(x, y) \cup \mathcal{B}$. The function $k(x, y, u, v)$ can be bounded as

$$|k(x, y, u, v)| \leq \frac{K_1}{(u-x)^2 + (v-y)^2} \left(1 + \frac{1}{r}((x - x_0)^2 + (y - y_0)^2)^{1/2}\right)^2.$$

Because $\mathcal{B}_{(x,y)} \subset \mathcal{T}$, we find that

$$|k(x, y, u, v)| \leq \frac{K_1}{(u-x)^2 + (v-y)^2} \left(1 + \frac{|\mathcal{T}|}{r}\right)^2.$$

Hence, the remainder can be estimated by

$$\begin{aligned}
|f - F_{d,\mathcal{B}} f| &\leq K_2 \sum_{\alpha+\beta=d+1} \int_{\mathcal{B}_{(x,y)}} \frac{|(x-u)^\alpha (y-v)^\beta|}{(u-x)^2 + (v-y)^2} \, |D_u^\alpha D_v^\beta f| \, du dv \\
&\leq K_2 \sum_{\alpha+\beta=d+1} \int_{\mathcal{T}} ((u-x)^2 + (v-y)^2)^{\frac{d-1}{2}} |D_u^\alpha D_v^\beta f| \, du dv,
\end{aligned}$$

with $K_2 = K_1(d+1)\left(1+\frac{|\mathcal{T}|}{r}\right)^2$. Let $1/p + 1/q = 1$, then Hölder's inequality for integrals and sums implies that

$$\begin{aligned}
|f - F_{d,\mathcal{B}} f| &\leq K_2 \sum_{\alpha+\beta=d+1} \left(\int_{\mathcal{T}} ((u-x)^2 + (v-y)^2)^{\frac{(d-1)}{2}q} \, dudv \right)^{1/q} \\
&\qquad \times \left(\int_{\mathcal{T}} |D_u^\alpha D_v^\beta f|^p \, dudv \right)^{1/p} \\
&\leq K_2 \left(\int_{\mathcal{T}} ((u-x)^2 + (v-y)^2)^{\frac{(d-1)}{2}q} \, dudv \right)^{1/q} \\
&\qquad \times \left(\sum_{\alpha+\beta=d+1} 1 \right)^{1/q} \left(\sum_{\alpha+\beta=d+1} \int_{\mathcal{T}} |D_u^\alpha D_v^\beta f|^p \, dudv \right)^{1/p} \\
&= K_2 \left((d+2) \int_{\mathcal{T}} ((u-x)^2 + (v-y)^2)^{\frac{(d-1)}{2}q} \, dudv \right)^{1/q} |f|_{d+1,p,\mathcal{T}}.
\end{aligned}$$

Assume that $(d-1)q + 2 > 0$, and consider the disk $\mathcal{D}$ with centre (x, y) and radius $|\mathcal{T}|$. It holds that $\mathcal{T} \subset \mathcal{D}$. We apply a transformation to polar coordinates to obtain that

$$\begin{aligned}
|f - F_{d,\mathcal{B}} f| &\leq K_2 \left((d+2) \int_{\mathcal{D}} ((u-x)^2 + (v-y)^2)^{\frac{(d-1)}{2}q} \, dudv \right)^{1/q} |f|_{d+1,p,\mathcal{T}} \\
&= K_2 \left(2\pi(d+2) \int_0^{|\mathcal{T}|} r^{(d-1)q+1} \, dr \right)^{1/q} |f|_{d+1,p,\mathcal{T}} \\
&= K_2 \left(\frac{2\pi(d+2)}{(d-1)q+2} \right)^{1/q} |\mathcal{T}|^{d-1+2/q} |f|_{d+1,p,\mathcal{T}} \\
&= K_1(d+1) \left(\frac{2\pi(d+2)}{(d-1)q+2} \right)^{1/q} \left(1 + \frac{|\mathcal{T}|}{r} \right)^2 |\mathcal{T}|^{d+1-2/p} |f|_{d+1,p,\mathcal{T}},
\end{aligned}$$

which results in (17). Note that for $p = 1$ ($q = \infty$) we get $K_t = K_1(d+1)$. It remains to check when $(d-1)q > -2$.

If $1 < p < \infty$, then $d - 1 > -2/q = 2/p - 2$ or $d > 2/p - 1$.
If $p = 1$, then $d - 1 \geq 0$ or $d \geq 1$.
If $p = \infty$, then $d > -1$ or $d \geq 0$. □

4 Local Approximation Error of QHPS Splines

Let the Cartesian coordinates of the corners $Q_{i,j}$ of the QHPS triangles t_i be denoted as $(X_{i,j}, Y_{i,j})$, and the Cartesian coordinates of conforming vertices V_i as (x_i, y_i). We suppose that all QHPS triangles are chosen as optimal triangles. We define the QHPS Hermite interpolant $\mathcal{I}_{QH} : C^1(\Omega) \to S^1_{2,H}(\Delta^*_H)$ as

$$\mathcal{I}_{QH} f = \sum_{i=1}^{n_c} \sum_{j=1}^{3} \psi_{i,j}(f) \, B^j_{i,QH}, \tag{18}$$

where $\psi_{i,j}(f)$ are linear functionals of the form

$$\psi_{i,j}(f) = f(V_i) + (X_{i,j} - x_i)D_x f(V_i) + (Y_{i,j} - y_i)D_y f(V_i). \tag{19}$$

Taking into account the following relation for a QHPS spline s_{QH} in representation (5) at a conforming vertex V_i,

$$\begin{aligned}
\begin{bmatrix} c_{i,1} \\ c_{i,2} \\ c_{i,3} \end{bmatrix} &= \begin{bmatrix} \alpha_{i,1} & \alpha_{i,2} & \alpha_{i,3} \\ \beta_{i,1} & \beta_{i,2} & \beta_{i,3} \\ \gamma_{i,1} & \gamma_{i,2} & \gamma_{i,3} \end{bmatrix}^{-1} \begin{bmatrix} s_{QH}(V_i) \\ D_x s_{QH}(V_i) \\ D_y s_{QH}(V_i) \end{bmatrix} \\
&= \begin{bmatrix} X_{i,1} & Y_{i,1} & 1 \\ X_{i,2} & Y_{i,2} & 1 \\ X_{i,3} & Y_{i,3} & 1 \end{bmatrix} \begin{bmatrix} x_i & y_i & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} s_{QH}(V_i) \\ D_x s_{QH}(V_i) \\ D_y s_{QH}(V_i) \end{bmatrix} \\
&= \begin{bmatrix} 1 & X_{i,1} - x_i & Y_{i,1} - y_i \\ 1 & X_{i,2} - x_i & Y_{i,2} - y_i \\ 1 & X_{i,3} - x_i & Y_{i,3} - y_i \end{bmatrix} \begin{bmatrix} s_{QH}(V_i) \\ D_x s_{QH}(V_i) \\ D_y s_{QH}(V_i) \end{bmatrix},
\end{aligned}$$

one can easily check that

$$\mathcal{I}_{QH}(s_{QH}) = s_{QH}, \qquad \forall s_{QH} \in S^1_{2,H}(\Delta^*_H), \tag{20}$$

and

$$\mathcal{I}_{QH} f(V_i) = f(V_i), \tag{21a}$$
$$D_x \mathcal{I}_{QH} f(V_i) = D_x f(V_i), \tag{21b}$$
$$D_y \mathcal{I}_{QH} f(V_i) = D_y f(V_i). \tag{21c}$$

We will use the Hermite interpolant $\mathcal{I}_{QH}$ for the characterization of the approximation power of QHPS splines. Given a sufficiently smooth function f, we prove in Theorem 11 that the QHPS spline $\mathcal{I}_{QH} f$ approximates the function f with an optimal order on each triangle in the hierarchical triangulation. This means that the approximation error measured on a triangle $\mathcal{T}$ is of the order $|\mathcal{T}|^3$.

We first introduce the mesh parameter D, called the level disparity.

Definition 8. *Let D_i be the largest difference between the level numbers of the triangles $\mathcal{T} \in \Delta_H$ that lie in the support $\mathcal{S}_i$ of the QHPS B-splines associated with a conforming vertex V_i. Define the level disparity D as the maximum of all constants D_i.*

In the worst case scenario, D equals the number of levels in the hierarchical triangulation. Yet, it is shown in [8] that a hierarchical triangulation can be locally adapted such that D is bounded by a constant independently of the number of levels in the hierarchical triangulation. Intuitively, one obtains a bounded level disparity when the level numbers of neighbouring triangles in Δ_H vary smoothly. A detailed procedure can be found in [8].

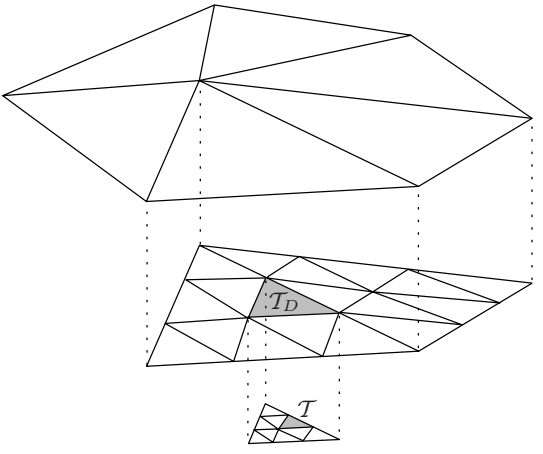


Fig. 2. The hierarchical structure $\mathbf{\Delta}_H$ of a given hierarchical triangulation Δ_H with three levels and a level disparity $D = 1$. A triangle $\mathcal{T} \in \Delta_H$ of level $l = 2$ and its corresponding triangle $\mathcal{T}_D \in \mathbf{\Delta}_H$ are shaded.

For a given triangle $\mathcal{T} \in \Delta_H$ with level number l, we denote triangle $\mathcal{T}_D$ as the triangle of level $\max\{l - D, 0\}$ in the hierarchical structure $\mathbf{\Delta}_H$ such that $\mathcal{T} \subseteq \mathcal{T}_D$. Figure 2 illustrates such a triangle for a hierarchical triangulation with three levels and $D = 1$.

We now formulate two lemmas with respect to length relations between the longest edges of particular triangles.

Lemma 9. *Suppose that $\mathcal{T}^* \in \Delta_H^*$ is a subtriangle of $\mathcal{T} \in \Delta_H$. The lengths of the longest edges of both triangles are related as*

$$|\mathcal{T}^*| \leq |\mathcal{T}| \leq K_{l^*} |\mathcal{T}^*|, \tag{22}$$

where $K_{l^*} = \sin(\theta_{\Delta_H^*})^{-2}$ with $\theta_{\Delta_H^*}$ the smallest angle in Δ_H^*.

Proof. The lower bound is trivial. To prove the upper bound, we consider the triangles $\mathcal{T}^*$ and $\mathcal{T}$ in Fig. 3. Let l_1 and l_2 be the lengths of two sides of the same triangle $\mathcal{T}$. By the law of sines it follows

$$\sin(\theta_{\mathcal{T}})\, l_1 \leq l_2 \leq |\mathcal{T}|, \tag{23}$$

with $\theta_{\mathcal{T}}$ the smallest angle in $\mathcal{T}$. By relation (23) it follows that the lengths in the figure are related as

$$l_{2,1} + l_{2,2} \leq \sin(\theta_{\Delta_H^*})^{-1} d_1. \tag{24}$$

Using relations (23) and (24), we obtain that

$$|\mathcal{T}| \leq \sin(\theta_{\Delta_H^*})^{-1}(l_{2,1}+l_{2,2}) \leq \sin(\theta_{\Delta_H^*})^{-2} d_1 \leq \sin(\theta_{\Delta_H^*})^{-2} |\mathcal{T}^*|. \qquad \square$$

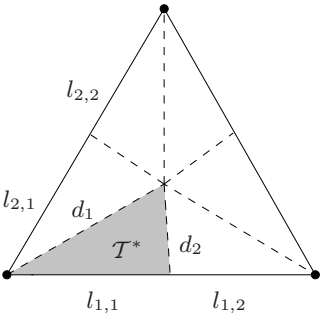


Fig. 3. A triangle $\mathcal{T}^*$ (shaded) in the HPS refinement Δ_H^*, inside the triangle $\mathcal{T} \in \Delta_H$

Lemma 10. *Suppose that $\mathcal{T}^{*,l}$ and $\mathcal{T}^{*,l+1}$ are triangles of a successive refinement level in $\mathbf{\Delta}_H^*$ such that $\mathcal{T}^{*,l+1} \subset \mathcal{T}^{*,l}$. The lengths of the longest edges of both triangles are related as*

$$|\mathcal{T}^{*,l+1}| \leq |\mathcal{T}^{*,l}| \leq K_{l^{**}} |\mathcal{T}^{*,l+1}|, \tag{25}$$

where $K_{l^{**}} = 5/4 \sin(\theta_{\Delta_H^*})^{-3}$ with $\theta_{\Delta_H^*}$ the smallest angle in Δ_H^*.

Proof. The lower bound is trivial. We consider the triangle $\mathcal{T}^{*,l}$ and the set of triangles $\mathcal{T}_m^{*,l+1}$ (with $m = 1, \ldots, 9$) in Fig. 4. Using the relation (23), we find that

$$\begin{aligned} |\mathcal{T}^{*,l}| &\leq \sin(\theta_{\Delta_H^*})^{-1} \left(1 + \sin(\theta_{\Delta_H^*})^{-2}\right) |\mathcal{T}_m^{*,l+1}|, \qquad \text{for} \quad m = 1, 6, 7, 9, \\ |\mathcal{T}^{*,l}| &\leq \sin(\theta_{\Delta_H^*})^{-1} \left(2 \sin(\theta_{\Delta_H^*})^{-1}\right) |\mathcal{T}_m^{*,l+1}|, \qquad \text{for} \quad m = 2, 3, 4, 5, 8. \end{aligned}$$

Since the minimal angle $\theta_{\Delta_H^*}$ in the HPS refinement can be bounded from above by $\pi/6$, we have

$$\begin{aligned} |\mathcal{T}^{*,l}| &\leq 5/4 \, \sin(\theta_{\Delta_H^*})^{-3} \, |\mathcal{T}_m^{*,l+1}|, \qquad \text{for} \quad m = 1, 6, 7, 9, \\ |\mathcal{T}^{*,l}| &\leq \sin(\theta_{\Delta_H^*})^{-3} \, |\mathcal{T}_m^{*,l+1}|, \qquad \text{for} \quad m = 2, 3, 4, 5, 8. \end{aligned}$$

This completes the proof. □

We finally come to the approximation theorem for QHPS splines. The next proof is inspired by the one for bivariate splines on conforming triangulations in [11,12].

Theorem 11 (Local approximation order). *Assume that Δ_H is a hierarchical triangulation of domain Ω with its level disparity D bounded independently of the number of levels in Δ_H, and assume that the corresponding HPS refinement Δ_H^* has a minimal angle $\theta_{\Delta_H^*}$ bounded from below by a positive constant. Let $\mathcal{T}^*$ be a triangle in Δ_H^*, and denote its level number by l. The triangle $\mathcal{T}_D$ is then defined as the triangle of level $\max\{l - D, 0\}$ in the hierarchical structure $\mathbf{\Delta}_H$ such that $\mathcal{T}^* \subseteq \mathcal{T}_D$. For any $f \in W_p^{k+1}(\Omega)$, the QHPS Hermite interpolant (18) locally behaves on $\mathcal{T}^*$ as*

$$\left\| D_x^\alpha D_y^\beta (f - \mathcal{I}_{QH} f) \right\|_{p,\mathcal{T}^*} \leq K_a |\mathcal{T}^*|^{k+1-\alpha-\beta} \, |f|_{k+1,p,\mathcal{T}_D}, \tag{26}$$

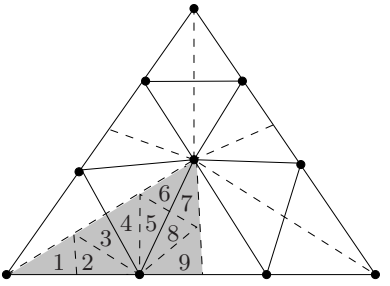


Fig. 4. A triangle $\mathcal{T}^{*,l}$ (shaded) and the nine smaller triangles $\mathcal{T}_m^{*,l+1}$ of a successive refinement level inside $\mathcal{T}^{*,l}$. The indices m are depicted in the corresponding triangles.

provided that $1 \leq k \leq 2$, $0 \leq \alpha + \beta \leq k$ *and* $2/k < p \leq \infty$. *The constant* K_a *is independent of* $|\mathcal{T}^*|$ *and* f.

Proof. We start by stating the following triangle inequality

$$\left\|D_x^\alpha D_y^\beta (f - \mathcal{I}_{QH} f)\right\|_{p,\mathcal{T}^*} \leq \left\|D_x^\alpha D_y^\beta (f - F_{k,\mathcal{B}_{\mathcal{T}^*}} f)\right\|_{p,\mathcal{T}^*} + \left\|D_x^\alpha D_y^\beta (\mathcal{I}_{QH} f - F_{k,\mathcal{B}_{\mathcal{T}^*}} f)\right\|_{p,\mathcal{T}^*},$$

where $F_{k,\mathcal{B}_{\mathcal{T}^*}} f$ is the averaged Taylor polynomial of degree k as defined in Lemma 6. The first term in the right-hand side can be bounded by (16), i.e.,

$$\|D_x^\alpha D_y^\beta (f - F_{k,\mathcal{B}_{\mathcal{T}^*}} f)\|_{p,\mathcal{T}^*} \leq K_{t,p} \, |\mathcal{T}^*|^{k+1-\alpha-\beta} |f|_{k+1,p,\mathcal{T}^*}. \tag{27}$$

We now examine the second term in more detail. The Hermite interpolant $\mathcal{I}_{QH}$ reproduces quadratic polynomials. Using (20) and (8)-(10), we obtain that

$$\begin{aligned}
\left\|D_x^\alpha D_y^\beta (\mathcal{I}_{QH} f - F_{k,\mathcal{B}_{\mathcal{T}^*}} f)\right\|_{p,\mathcal{T}^*} &= \left\|D_x^\alpha D_y^\beta \mathcal{I}_{QH} (f - F_{k,\mathcal{B}_{\mathcal{T}^*}} f)\right\|_{p,\mathcal{T}^*} \\
&\leq \left(\frac{6}{r_{\mathcal{T}^*}}\right)^{\alpha+\beta} {\mathcal{A}_{\mathcal{T}^*}}^{1/p} \left\|\mathcal{I}_{QH} (f - F_{k,\mathcal{B}_{\mathcal{T}^*}} f)\right\|_{\infty,\mathcal{T}^*} \\
&\leq \left(\frac{6}{r_{\mathcal{T}^*}}\right)^{\alpha+\beta} 12^{3/p} \left\|\mathcal{I}_{QH} (f - F_{k,\mathcal{B}_{\mathcal{T}^*}} f)\right\|_{p,\mathcal{T}^*} \\
&= \frac{K_3}{{r_{\mathcal{T}^*}}^{\alpha+\beta}} \left\|\mathcal{I}_{QH} (f - F_{k,\mathcal{B}_{\mathcal{T}^*}} f)\right\|_{p,\mathcal{T}^*}.
\end{aligned} \tag{28}$$

Fix (x, y) in the triangle $\mathcal{T}^*$. Since the QHPS basis forms a convex partition of unity, we can bound the Hermite interpolant $\mathcal{I}_{QH}$ in (18)-(19) as

$$\begin{aligned}
\left|\mathcal{I}_{QH} f(x,y)\right| &\leq \sum_{i|_{\mathcal{T}^* \subseteq \mathcal{S}_i}} \sum_j |\psi_{i,j}(f)| \, B_{i,QH}^j(x,y) \\
&\leq \max_{i|_{\mathcal{T}^* \subseteq \mathcal{S}_i}, j} |f(V_i) + (X_{i,j} - x_i) D_x f(V_i) + (Y_{i,j} - y_i) D_y f(V_i)|.
\end{aligned}$$

By the definitions of the level disparity D and the triangle $\mathcal{T}_D$, we have

$$\begin{aligned}\left|\mathcal{I}_{QH} f\right| \leq \|f\|_{\infty,\mathcal{T}_D} &+ \max_{i|_{\mathcal{T}^*\subseteq\mathcal{S}_i},j} |X_{i,j}-x_i|\,\|D_x f\|_{\infty,\mathcal{T}_D} \\ &+ \max_{i|_{\mathcal{T}^*\subseteq\mathcal{S}_i},j} |\,Y_{i,j}-y_i\,|\,\|D_y f\|_{\infty,\mathcal{T}_D}.\end{aligned}$$

We know from [6, 8] that the length of the longest edge of an optimal QHPS triangle can be bounded as $|t_i| \leq K_4\,|\mathcal{T}_D|$, where K_4 only depends on the minimal angle in Δ_H^*. Since

$$|X_{i,j}-x_i| \leq |t_i|, \quad |\,Y_{i,j}-y_i\,| \leq |t_i|,$$

it follows that

$$\begin{aligned}\left|\mathcal{I}_{QH}(f-F_{k,\mathcal{B}_{\mathcal{T}^*}}f)\right| \leq \|f-F_{k,\mathcal{B}_{\mathcal{T}^*}}f\|_{\infty,\mathcal{T}_D} &\\ + K_4\,|\mathcal{T}_D|\,\|D_x(f-F_{k,\mathcal{B}_{\mathcal{T}^*}}f)\|_{\infty,\mathcal{T}_D} &\\ + K_4\,|\mathcal{T}_D|\,\|D_y(f-F_{k,\mathcal{B}_{\mathcal{T}^*}}f)\|_{\infty,\mathcal{T}_D},&\end{aligned}$$

and using Lemma 5, we get

$$\begin{aligned}\left|\mathcal{I}_{QH}(f-F_{k,\mathcal{B}_{\mathcal{T}^*}}f)\right| \leq \|f-F_{k,\mathcal{B}_{\mathcal{T}^*}}f\|_{\infty,\mathcal{T}_D} &\\ + K_4\,|\mathcal{T}_D|\,\|D_x f-F_{k-1,\mathcal{B}_{\mathcal{T}^*}}(D_x f)\|_{\infty,\mathcal{T}_D} &\\ + K_4\,|\mathcal{T}_D|\,\|D_y f-F_{k-1,\mathcal{B}_{\mathcal{T}^*}}(D_y f)\|_{\infty,\mathcal{T}_D}.&\end{aligned}$$

By Lemma 7 we find for $2/k < p \leq \infty$ that

$$\left|\mathcal{I}_{QH}(f-F_{k,\mathcal{B}_{\mathcal{T}^*}}f)\right| \leq K_t(1+2\,K_4)\left(1+\frac{|\mathcal{T}_D|}{r_{\mathcal{T}^*}}\right)^2 |\mathcal{T}_D|^{k+1-2/p}|f|_{k+1,p,\mathcal{T}_D}.$$

From Lemmas 9 and 10 we get $|\mathcal{T}_D| \leq K_{l^*}(K_{l^{**}})^D|\mathcal{T}^*|$. Filling this inequality and the following bound for the inradius [6],

$$r_{\mathcal{T}} \geq \tan\left(\frac{\theta_{\mathcal{T}}}{2}\right)\frac{|\mathcal{T}|}{2}, \tag{29}$$

into the upper bound for $\left|\mathcal{I}_{QH}(f-F_{k,B_{\mathcal{T}^*}}f)\right|$, we arrive at

$$\left|\mathcal{I}_{QH}(f-F_{k,\mathcal{B}_{\mathcal{T}^*}}f)\right| \leq K_5\,|\mathcal{T}^*|^{k+1-2/p}|f|_{k+1,p,\mathcal{T}_D},$$

with K_5 independent of $|\mathcal{T}^*|$ and f. Hence,

$$\begin{aligned}\left\|\mathcal{I}_{QH}(f-F_{k,\mathcal{B}_{\mathcal{T}^*}}f)\right\|_{p,\mathcal{T}^*} &= \left(\int_{\mathcal{T}^*}|\mathcal{I}_{QH}(f-F_{k,\mathcal{B}_{\mathcal{T}^*}}f)|^p\right)^{1/p} \\ &\leq K_5\,|\mathcal{T}^*|^{k+1-2/p}{\mathcal{A}_{\mathcal{T}^*}}^{1/p}|f|_{k+1,p,\mathcal{T}_D} \\ &\leq K_5\,|\mathcal{T}^*|^{k+1}|f|_{k+1,p,\mathcal{T}_D}. \end{aligned} \tag{30}$$

Combining inequalities (27)-(30), we obtain approximation bound (26) with

$$K_a = K_{t,p} + K_3K_5\big(2/\tan(\theta_{\Delta_H^*}/2)\big)^{\alpha+\beta}. \qquad \square$$

Remark that Theorem 11 is a generalization of the approximation theorem in [13] for classical Powell-Sabin splines. There, a related upper bound is provided for the error of a Powell-Sabin spline interpolant analogous to (18)-(19).

5 Numerical Example

We illustrate the derived approximation order of QHPS splines with a numerical example. We consider the test function

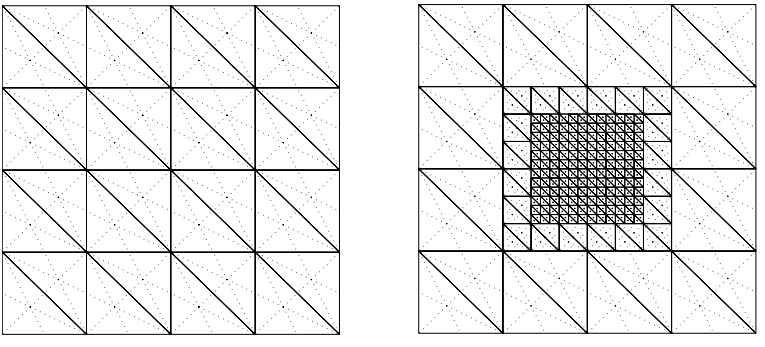
$$f(x, y) = \exp(-16(x^2 + y^2)) \tag{31}$$

on the domain $[-1, 1] \times [-1, 1]$. We compute the Hermite interpolating QHPS spline $\mathcal{I}_{QH} f$ on a nested sequence of locally refined hierarchical triangulations. The initial triangulation is a regular conforming mesh, consisting of 32 triangles, as shown in Fig. 5(a). Define a triangulation patch $\mathcal{P} \subseteq \Delta_H$ as the set of all triangles on the finest level in hierarchical triangulation Δ_H. We obtain a set of successive meshes by refining triadically each triangle in the corresponding $\mathcal{P}$ that is not situated at the boundary of $\mathcal{P}$. Figure 5(b) depicts such a hierarchical triangulation after two refinement steps. One can check that its level disparity equals $D = 1$.

Figure 6 shows the reduction of the (local) interpolation error. This error is computed on the triangulation patches of the successive hierarchical triangulations. It is measured by the L_∞-norm $\left\| f - \mathcal{I}_{QH} f \right\|_{\infty,\mathcal{P}}$ and by

$$\frac{\left\| f - \mathcal{I}_{QH} f \right\|_{2,\mathcal{P}}}{\sqrt{\mathcal{A}_\mathcal{P}}}, \tag{32}$$

with $\mathcal{A}_\mathcal{P}$ the area of patch $\mathcal{P}$. Since the considered triangulation patches have a different area, a fair comparison directs us to use an L_2-norm scaled by the square root of the area of the patch. As expected, the error behaves as $\mathcal{O}(|\mathcal{P}|^3)$.



(a) (b)

Fig. 5. (a) Initial triangulation with its HPS refinement indicated in dotted lines. (b) Locally refined hierarchical triangulation with three levels. The level disparity in the triangulation equals $D = 1$.

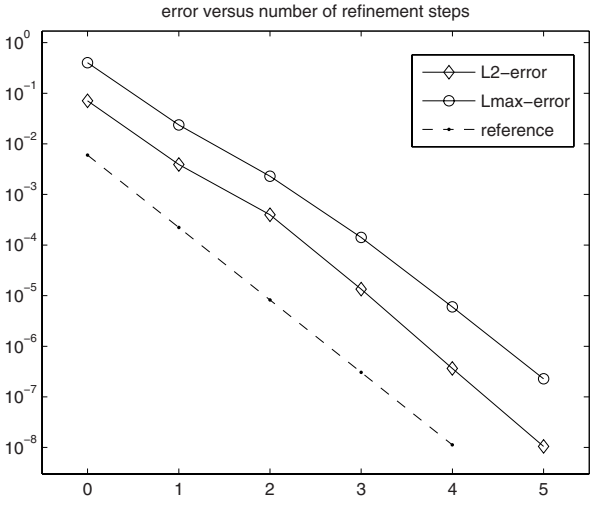


Fig. 6. Convergence of the error, measured by the L_∞-norm and by (32), of the Hermite interpolating QHPS spline $\mathcal{I}_{QH} f$, computed for test function (31) on the triangulation patches $\mathcal{P}$ of a nested sequence of locally refined hierarchical triangulations. The reference direction shows $\mathcal{O}(|\mathcal{P}|^3)$.

6 Conclusion

We have proved that the local approximation order of QHPS splines is optimal on particularly constructed hierarchical triangulations. For a sufficiently smooth function f, there exists a Hermite interpolating QHPS spline $\mathcal{I}_{QH} f$, where the approximation error measured on a triangle $\mathcal{T}^*$ in the HPS refinement is of the order $|\mathcal{T}^*|^3$.

Acknowledgement

Hendrik Speleers is a Postdoctoral Fellow of the Research Foundation Flanders (Belgium).

References

1. Powell, M., Sabin, M.: Piecewise quadratic approximations on triangles. ACM Trans. Math. Softw. 3, 316–325 (1977)
2. Dierckx, P.: On calculating normalized Powell-Sabin B-splines. Comput. Aided Geom. Design 15, 61–78 (1997)
3. He, Y., Jin, M., Gu, X., Qin, H.: A C^1 globally interpolatory spline of arbitrary topology. In: Paragios, N., Faugeras, O., Chan, T., Schnörr, C. (eds.) Proc. 3rd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision, Beijing, China, pp. 295–306 (2005)

4. Dierckx, P., Van Leemput, S., Vermeire, T.: Algorithms for surface fitting using Powell-Sabin splines. IMA J. Numer. Anal. 12, 271–299 (1992)
5. Manni, C., Sablonnière, P.: Quadratic spline quasi-interpolants on Powell-Sabin partitions. Adv. Comput. Math. 26, 283–304 (2007)
6. Speleers, H., Dierckx, P., Vandewalle, S.: Quasi-hierarchical Powell-Sabin B-splines. Comput. Aided Geom. Design 26, 174–191 (2009)
7. Grinspun, E., Krysl, P., Schröder, P.: CHARMS: a simple framework for adaptive simulation. ACM Trans. Graphics 21, 281–290 (2002)
8. Speleers, H., Dierckx, P., Vandewalle, S.: On the L_p-stability of quasi-hierarchical Powell-Sabin B-splines. In: Neamtu, M., Schumaker, L. (eds.) Approximation Theory XII: San Antonio 2007, pp. 398–413. Nashboro Press (2008)
9. Kroó, A., Révész, S.: On Bernstein and Markov-type inequalities for multivariate polynomials on convex bodies. J. Approx. Theory 99, 134–152 (1999)
10. Brenner, S., Scott, L.: The mathematical theory of finite element methods, 2nd edn. Springer, Heidelberg (2002)
11. Lai, M., Schumaker, L.: On the approximation power of bivariate splines. Adv. Comput. Math. 9, 251–279 (1998)
12. Lai, M., Schumaker, L.: Spline Functions on Triangulations. Encyclopedia of Mathematics and its Applications, vol. 110. Cambridge University Press, Cambridge (2007)
13. Maes, J., Bultheel, A.: Surface compression with hierarchical Powell-Sabin B-splines. Int. J. Wav. Multires. Inf. Proc. 4, 177–196 (2006)

Logarithmic Curvature and Torsion Graphs

Norimasa Yoshida[1], Ryo Fukuda[2], and Takafumi Saito[3]

[1] Nihon University, Chiba 275-8575, Japan
norimasa@acm.org
[2] Nihon University, Chiba 275-8575, Japan
[3] Tokyo University of Agriculture and Technology
Tokyo 185-8588, Japan
txsaito@cc.tuat.ac.jp

Abstract. This paper introduces logarithmic curvature and torsion graphs for analyzing planar and space curves. We present a method for drawing these graphs from any differentiable parametric curves and clarify the characteristics of these graphs. We show several examples of theses graphs drawn from planar and 3D Bézier curves. From the graphs, we can see some interesting properties of curves that cannot be derived from the curvature or torsion plots.

1 Introduction

To design high quality shapes, CAD systems are usually equipped with tools for inspecting the geometric quality of curves and surfaces. For curves, the curvature plot, which is the plot of curvature versus arc length or versus the parameter, is commonly used [1]. For surfaces, the color coding of Gaussian curvature or mean curvature, the reflection lines [8], or isophotes [10] are used. This paper proposes a new method for identifying the shape information of planar and space curves. For planar curves, we propose to use logarithmic curvature graphs (LCGs). For space curves, we propose to use logarithmic curvature graphs and logarithmic torsion graphs (LTGs). Note that the term "logarithmic curvature" has been used by Farouki et al. in [3]. We use the term in an entirely different context.

A logarithmic curvature graph is a graph whose horizontal and vertical axes are $\log\rho$ and $\log\left(\rho\frac{\mathrm{d}s}{\mathrm{d}\rho}\right)$ [9, 11], where ρ and s are the radius of curvature and the arc length, respectively. An LCG is the analytical representation of a logarithmic curvature histogram(LCH) originally proposed by Harada [6, 7]. Harada has used LCHs to analyze curvature variation properties of curves in existing aesthetic objects and discovered that LCHs (i.e. LCGs) of many such curves become linear. The overall shapes of curves whose LCG is linear has been clarified and a method for drawing a curve segment has been proposed in [11]. If the LCG of a curve is linear, its slope α is known as the type of the curve [6]. If $\alpha = -1, 0, 1$ or 2, the curve is known to be the Clothoid, the Nielsen's spiral [5], the logarithmic spiral, or the circle involute [6, 11]. Conversely, if we could draw the LCG of a planar curve and found that the LCG is almost linear with slope α, we can find the type of the curve. If α is close to -1, it means that the curve is close to the Clothoid. To identify the shape information of space curves, we need to append

M. Dæhlen et al. (Eds.): MMCS 2008, LNCS 5862, pp. 434–443, 2010.

LTGs onto LCGs. The role of torsion variation is not clearly known for aesthetic space curves. However, for a certain class of space curves, if we could know both the LCGs and LTGs are linear, we could get the shape information of the curve. LCGs and LTGs can be used to find the shape information of curves.

In this paper, we propose LCGs and LTGs to be drawn from any differential parametric curves, such as Bézier curves, or NURBS curves. We clarify the characteristics of LCGs and LTGs and show several examples of theses graphs drawn from Bézier curves. LCGs and LTGs do not replace curvature and torsion plots. LCGs and LTGs offer the information related to the curvature and torsion variations that the curvature and torsion plots do not offer. For planar curves, LCGs are already introduced in [12, 13]. In this paper, however, we draw LCGs and LTGs from space curves and derive more detailed characteristics. Moreover, using the LCGs, we show that we can find an interesting property of polynomial curves near the inflection point. Using LCGs and LTGs, we show that the slopes of LCGs and LTGs of typical 3D class A Bézier curves [2] get closer to 1 as the degree gets higher.

2 Logarithmic Curvature Graphs and Their Characteristics

2.1 Logarithmic Curvature Graphs

LCGs are defined to be graphs whose horizontal and vertical axes are given by $\log\rho$ and $\log\left(\rho\left|\frac{\mathrm{d}s}{\mathrm{d}\rho}\right|\right)$, respectively. See Fig. 1(a). Comparing the previous definition of LCGs, which was derived from the linearity of LCG [9, 11]

$$\log\left(\rho\frac{\mathrm{d}s}{\mathrm{d}\rho}\right) = \alpha\log\rho + c \ , \tag{1}$$

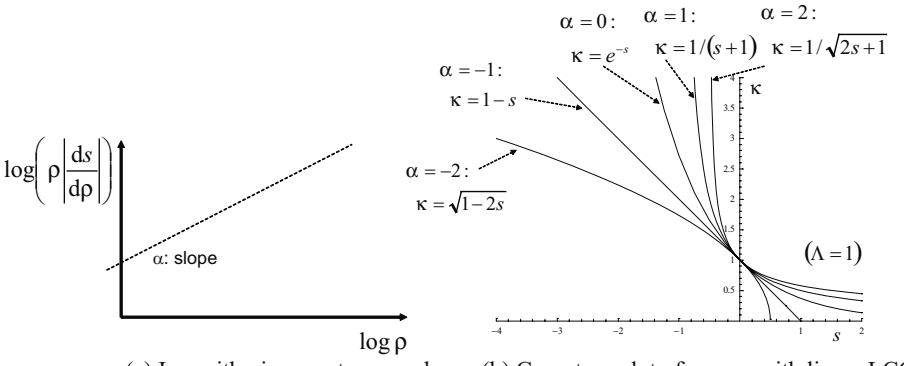
the absolute value is taken for $\frac{\mathrm{d}s}{\mathrm{d}\rho}$. This is because, in the previous definition, ρ is assumed to be monotonically increasing with respect to s. The effect of taking the absolute value can be considered as reparameterizing the curve so that the radius of curvature becomes monotonically increasing with respect to the arc length.

Let $\mathbf{x}(t)$ be a differentiable curve with parameter t. We represent the first, second and third derivatives of $\mathbf{x}(t)$ with respect to t by $\dot{\mathbf{x}}$, $\ddot{\mathbf{x}}$ and $\dddot{\mathbf{x}}$, respectively. To draw the LCG, we need to compute both $\log\rho$ and $\log\left(\rho\left|\frac{\mathrm{d}s}{\mathrm{d}\rho}\right|\right)$. We can easily compute $\log\rho$. Let κ be the curvature. We can compute $\log\left(\rho\left|\frac{\mathrm{d}s}{\mathrm{d}\rho}\right|\right)$ using the following equations:

$$\frac{\mathrm{d}\rho}{\mathrm{d}s} = -\frac{1}{\kappa^2}\frac{\mathrm{d}\kappa}{\mathrm{d}s} \ , \tag{2}$$

$$\frac{\mathrm{d}\kappa}{\mathrm{d}s} = \frac{(\dot{\mathbf{x}}\wedge\dddot{\mathbf{x}})\cdot(\dot{\mathbf{x}}\wedge\ddot{\mathbf{x}})(\dot{\mathbf{x}}\cdot\dot{\mathbf{x}}) - 3|\dot{\mathbf{x}}\wedge\ddot{\mathbf{x}}|^2(\dot{\mathbf{x}}\cdot\ddot{\mathbf{x}})}{|\dot{\mathbf{x}}|^6\ |\dot{\mathbf{x}}\wedge\ddot{\mathbf{x}}|} \ . \tag{3}$$

Note that (3) works for both planar and space curves. For planar curves, however, it is more efficient to use the simplified version of $\frac{\mathrm{d}\kappa}{\mathrm{d}s}$ [1, 11].



(a) Logarithmic curvature graph (b) Curvature plot of curves with linear LCGs

Fig. 1. Logarithmic curvature graph and the curvature plot of curves with linear LCGs

At the point of $\frac{\mathbf{d}\rho}{\mathbf{d}s} = 0$ (or $\frac{\mathrm{d}\kappa}{\mathrm{d}s} = 0$), the vertical value of the LCG is not defined. As the curve gets close to the point of $\frac{\mathbf{d}\rho}{\mathbf{d}s} = 0$, the vertical value gets close to infinity since ρ and $|\frac{\mathrm{d}s}{\mathrm{d}\rho}|$ get close to a constant and infinity, respectively. Therefore, when we draw the LCG of a curve with not monotonically varying curvature, the vertical values of the LCG contain the same number of infinite values as the number of curvature extrema. Both the vertical and the horizontal values of the LCG is not well-defined at the point of $\rho = 0$ or $\rho = \infty$. In [11], we have clarified that a curve with linear LCG whose slope is greater than or equal to zero has an inflection point at infinity. A curve with linear LCG whose slope is smaller than 0 has an inflection point (not at infinity). Corresponding to these properties, if the radius of curvature of a curve approaches infinity as the arc length approaches infinity, both the horizontal and vertical values of the LCG approaches ∞. If the LCG is linear, this means that the slope of the LCG is greater than or equal to 0. If the radius of curvature of a curve becomes infinite not at infinity, the horizontal value of the LCG becomes ∞ whereas the vertical value becomes $-\infty$ at that point. Again, if the LCG is linear, this means that the slope of the LCG is smaller than 0.

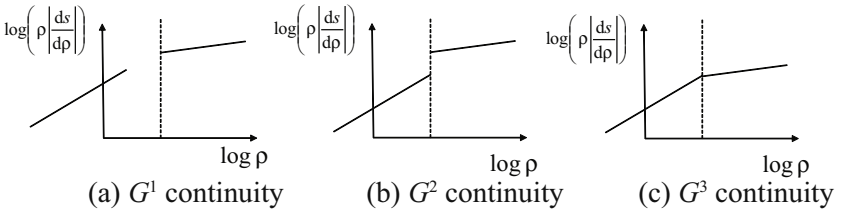
Modifying (1), we get the curvature function with respect to the arc length [11]:

$$\kappa = \begin{cases} e^{-\Lambda s} & if\ \alpha = 0 \\ (\Lambda\alpha s + 1)^{-\frac{1}{\alpha}} & otherwise \end{cases} \tag{4}$$

where $\Lambda = e^{-c}$. See Fig.1(b). If the LCG is (approximately) linear, the curvature function can be (approximately) represented by (4).

2.2 Continuity of Curves and Continuity of LCGs

We consider the continuity of a curve and the continuity of its LCG. If a curve is G^1 (or less than G^1) continuous at some point of the curve, both the horizontal and vertical values of the LCG becomes different at the point. If a curve is G^2 continuous at some



(a) G^1 continuity (b) G^2 continuity (c) G^3 continuity

Fig. 2. LCGs of curves with G^1, G^2 and G^3 continuity

point, the horizontal values of the LCG becomes the same. But the vertical values become different at the point. If a curve is G^3 continuous at some point of the curve, the LCG of the curve becomes positionally(G^0) continuous since both the horizontal values and vertical values take the same values. See Fig.2 for the continuity of curves and the continuity of LCGs. Note that the positional continuity of the LCG does not necessarily mean that the curve is G^3 continuous. Suppose there are two curve segments with G^3 continuity. Translating or rotating one of the two segments does not change the LCG since it does not affect the differential properties of curves.

2.3 Formulation by Curvature

LCGs in the above sections are formulated by the radius of curvature ρ. LCGs can also be formulated by curvature κ. Except for this section, we say LCGs to mean LCGs formulated by the radius of curvature.

We simply consider replacing ρ of LCGs formulated by the radius of curvature by κ. Then the horizontal and vertical values of LCGs formulated by curvature are $\log\kappa$ and $\log\left(\kappa\left|\frac{ds}{d\kappa}\right|\right)$, respectively. Since

$$\log\rho = -\log\kappa \ , \tag{5}$$

horizontal values of LCGs formulated by the radius of curvature and LCGs formulated by curvature have the same absolute value but opposite signs.

Using (2) and $\kappa = \frac{1}{\rho}$,

$$\rho\frac{ds}{d\rho} = -\kappa\frac{ds}{d\rho} \ . \tag{6}$$

Since ρ and κ are both positive, taking the absolute values of $\frac{ds}{d\rho}$ and $\frac{ds}{d\kappa}$, we get

$$\rho\left|\frac{ds}{d\rho}\right| = \kappa\left|\frac{ds}{d\rho}\right| \ . \tag{7}$$

Therefore, the vertical values of LCGs formulated by the radius of curvature and the vertical values of LCGs formulated by curvature are the same. Thus the LCG of a curve formulated by the radius of curvature is symmetric to the LCG formulated by curvature with respect to the vertical axis of $\log\rho = 0$.

3 Logarithmic Torsion Graphs and Their Characteristics

3.1 Logarithmic Torsion Graphs

We define logarithmic torsion graphs by replacing ρ of LCGs by the radius of torsion μ. The linearity of the LTG can be represented by

$$\log\left(\mu\frac{\mathrm{d}s}{\mathrm{d}\mu}\right) = \beta\log\mu + d \ , \tag{8}$$

where β is the slope of the straight line and d is a constant. See Fig.3 for the LTG. Similar properties as in LCGs hold for LTGs. Thus we try to abbreviate the explanation as much as possible.

Similarly as in Section 2.1, we represent a curve by $\mathbf{x}(t)$ and denote the fourth derivative of $\mathbf{x}(t)$ with respect to t by $\dddot{\mathbf{x}}$. Let τ be the torsion. Then the LTG of a curve can be drawn using the following equations:

$$\frac{\mathrm{d}\mu}{\mathrm{d}s} = -\frac{1}{\tau^2}\frac{\mathrm{d}\tau}{\mathrm{d}s} \ , \tag{9}$$

$$\frac{\mathrm{d}\tau}{\mathrm{d}s} = \frac{(\dot{\mathbf{x}}\wedge\ddot{\mathbf{x}})\cdot\dddot{\mathbf{x}}\,|\dot{\mathbf{x}}\wedge\ddot{\mathbf{x}}|^2 - 2\,((\dot{\mathbf{x}}\wedge\ddot{\mathbf{x}})\cdot\dddot{\mathbf{x}})\,(\dot{\mathbf{x}}\wedge\dddot{\mathbf{x}})\cdot(\dot{\mathbf{x}}\wedge\ddot{\mathbf{x}})}{|\dot{\mathbf{x}}\wedge\ddot{\mathbf{x}}|^4|\dot{\mathbf{x}}\cdot\dot{\mathbf{x}}|^{\frac{1}{2}}} \ . \tag{10}$$

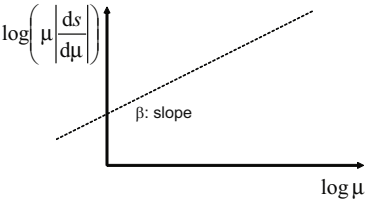


Fig. 3. Logarithmic Torsion Graph

At the point of $\frac{\mathrm{d}\mu}{\mathrm{d}s} = 0$ (or $\frac{\mathrm{d}\tau}{\mathrm{d}s} = 0$), the vertical value of LTG becomes infinite. Therefore at the torsion extremum, the vertical value of the LTG becomes infinite. At the point of $\mu = 0$ or $\mu = \infty$, both the horizontal and vertical values of the LTG are not defined. Similarly as in LCG, as the curve approaches the point at $\mu = \infty$ in infinite distance, both the horizontal and vertical values of LTG approaches ∞. If the point at $\mu = \infty$ is in a finite distance, the horizontal value of LTG approaches ∞ whereas the vertical value approaches ∞, as the curve approaches the point at $\mu = \infty$.

Modifying (8), we can obtain the torsion function with respect to the arc length:

$$\tau = \begin{cases} e^{-\Omega s} & if\ \beta = 0 \\ (\Omega\beta s + 1)^{-\frac{1}{\beta}} & otherwise \end{cases} \tag{11}$$

where $\Omega = e^{-d}$. The (approximate) linearity of LTG means that the torsion function is (approximately) represented by (11).

3.2 Continuity of Curves and Continuity of LTGs

The torsion is related to the third derivative of a curve. See Fig.4 to see the relationship between the continuity of curves and the continuity of the LTG.

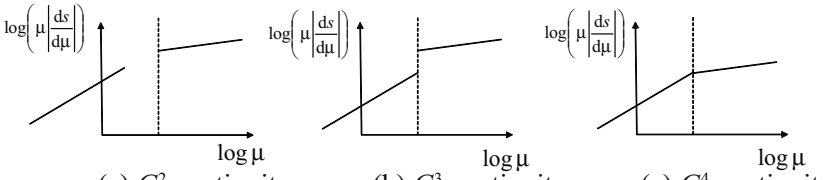


Fig. 4. LTGs of curves with G^2, G^3 and G^4 continuity

3.3 Formulation by Torsion

Since

$$\log\mu = -\log\tau, \quad \log\mu\left|\frac{ds}{d\mu}\right| = \log\tau\left|\frac{ds}{d\tau}\right| \quad , \tag{12}$$

the LTG formulated by the torsion is symmetric to the LTG formulated by the radius of torsion with respect to the axis $\log\mu = 0$.

4 Examples of LCGs and LTGs

4.1 Planar Bézier Curves

Fig.5 shows the curve, the curvature plot, and the LCGs of three planar Bézier curves. Fig.5(a) is a Bézier curve with not monotonically varying curvature. There is a point in the LCG where the vertical value becomes infinite since there is a curvature extremum in the curve. Fig.5(b) is a Bézier curve with monotonically varying curvature. Although the curvature is monotonically varying, its LCG is not linear. Therefore this Bézier curve is not very aesthetically pleasing from the viewpoint of Harada's analysis [6,7]. Fig.5(c) is a Bézier curve with almost linear LCG. The slope of the LCG implies the type(α) of log-aesthetic curves.

Using LCGs, we can find an interesting property of polynomial curves. Fig.6 (a) and (b) are cubic and quintic Bézier curves with inflection points. As the curve approaches the inflection point (in a finite distance), the horizontal value ($\log\rho$) of the LCG approaches ∞ whereas the vertical value $\left(\log\left(\rho\left|\frac{ds}{d\rho}\right|\right)\right)$ approaches $-\infty$. In both of the LCGs, we can see that the slopes approach -1 as the curve approaches the inflection point. This indicates that in the vicinity of the inflection point, the curve can be approximated by the Clothoid(the log-aesthetic curve with $\alpha = -1$). We found that in any polynomial Bézier curves of any degree, the slope of the LCG near the inflection point is -1 except when the control points are placed in specific positions. Fig.6(c) and (d) show the cases where the slopes of the LCGs are not -1 near the inflection points. The slopes of the LCGs near the inflection point of Fig.6(c) and (d) are $\frac{1}{2}$ and $\frac{1}{4}$. This

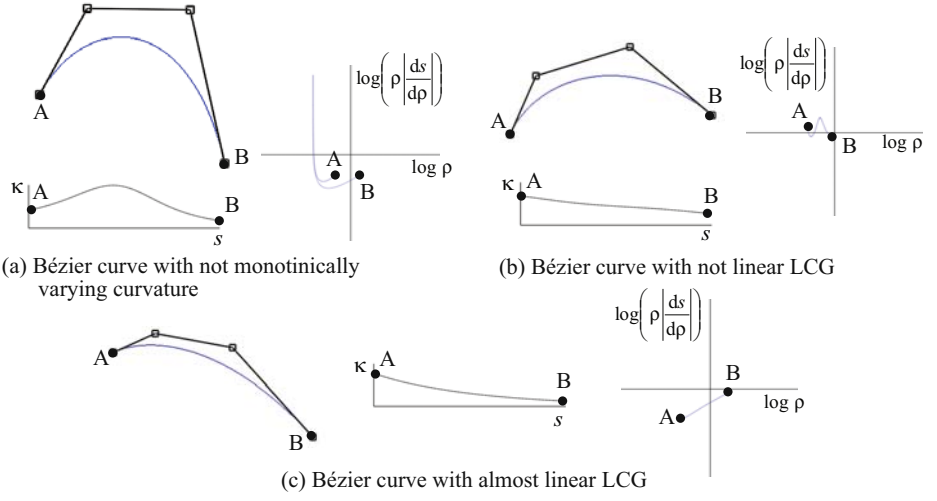


Fig. 5. Curvature plots and LCGs of various Bézier curves

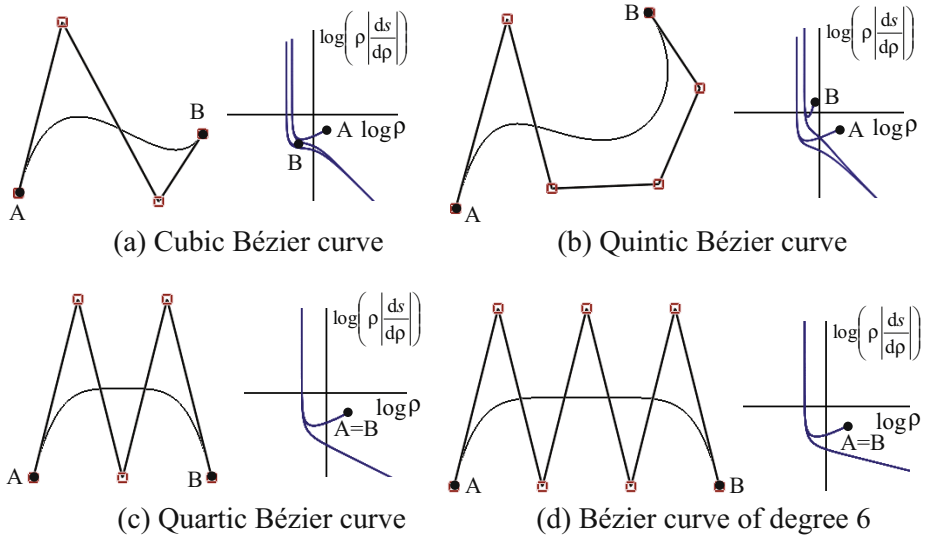


Fig. 6. LCGs of Bézier curves with inflection points

indicates if the control points of a Bézier curve of degree n are placed in a specific position as shown in Fig. 6 (c) or (d), the curve can be approximated, near the inflection point, by log-aesthetic curves with $\alpha = -\frac{1}{n-2}$. Note that LCGs do not give proofs for the properties of curves. They provide the shape information of curves for further investigation.

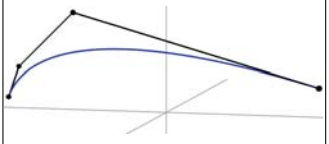
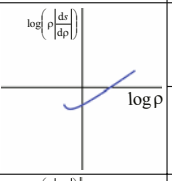
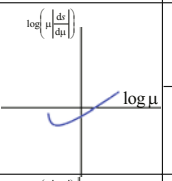
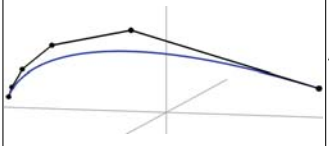
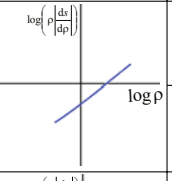
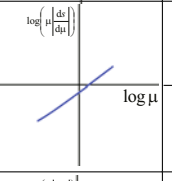
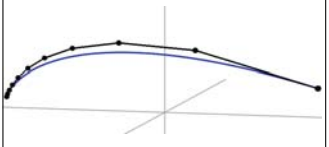
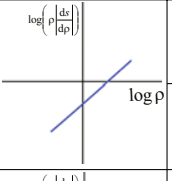
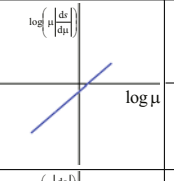
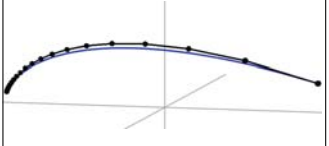
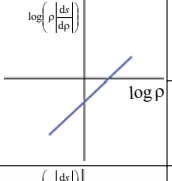
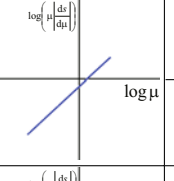
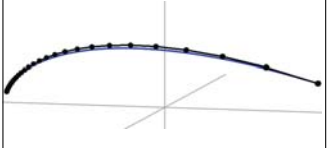
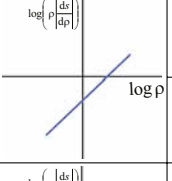
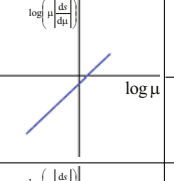
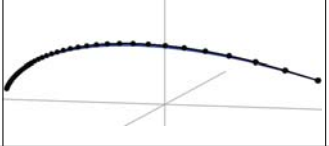
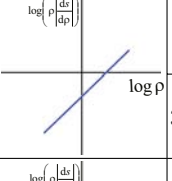
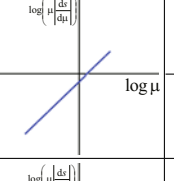
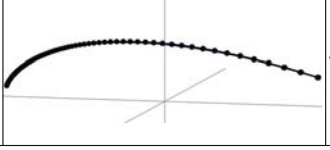
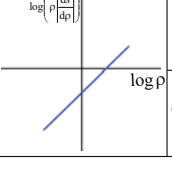
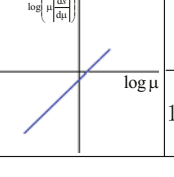
deg.	curves	LCG	slope var	LTG	slope var
3			0. 574 12. 15		0. 453 37. 12
5			0. 780 0. 425		0. 732 0. 459
10			0. 892 0. 096		0. 871 0. 118
20			0. 945 0. 026		0. 936 0. 054
30			0. 963 0. 003		0. 957 0. 011
50			0. 978 $3.0 \cdot 10^{-4}$		0. 974 0. 001
99			0. 990 $7.2 \cdot 10^{-5}$		0. 987 $1.7 \cdot 10^{-4}$

Fig. 7. LCGs and LTGs of typical 3D class A Bézier curves

4.2 3D Bézier Curves

As an application of LCGs and LTGs, we show that these graphs can be used to find the shape information of space curves. We found that the LCGs and LTGs of typical 3D class A Bézier curves [2] get closer to be linear with slopes both 1, as the degree gets higher. This means that typical 3D class A Bézier curves get closer to 3D logarithmic spirals as the degree gets higher.

Let n and $\mathbf{b}_i (0 \leq i \leq n)$ be the degree and the control points of a Bézier curve, respectively. Let $\Delta \mathbf{b}_i = \mathbf{b}_{i+1} - \mathbf{b}_i (0 \leq i < n)$ be the forward difference of control point vectors. Typical 3D class A Bézier curves are curves with monotonically varying curvature and torsion having the relationship

$$\Delta \mathbf{b}_i = M^i \Delta \mathbf{b}_0 \quad (1 \leq i < n) \tag{13}$$

where M is a matrix composed of a rotation about an axis and a scaling. Note that (13) does not guarantee the curvature and torsion are monotonically varying. A method for drawing typical 3D class A Bézier curves by specifying two endpoints and their tangents is proposed in [4]. Drawing the LCGs and LTGs for typical 3D class A Bézier curves, we found that both LCGs and LTGs get closer to be linear with slopes 1 when the degree gets higher. Fig. 7 is an example where the degree is changed with the same endpoint constraints. In the figure, the slopes are computed by the least square method and "var." means the variance. A curve whose slopes of the LCG and LTG are both 1 is the 3D extension of a logarithmic spiral:

$$L(t) = \left(e^{at} \cos t, e^{at} \sin t, be^t\right), \tag{14}$$

where a and b are constants. LCGs and LTGs do not give a proof that typical 3D class A Bézier curves are approximations to the 3D extensions of a logarithmic spiral. However, they give the shape information concerning the curves, which the curvature and torsion plots do not offer.

5 Conclusions

We have proposed logarithmic curvature graphs (LCGs) and logarithmic torsion graphs (LTGs) for analyzing the characteristics and identifying the shape information of any differentiable parametric curves. We have clarified the characteristics of these graphs and showed several examples of the graphs drawn from planar and space Bézier curves. Using LCGs and LTGs, we showed that these graphs can identify the shape information of curves, which the curvature and torsion plots do not offer. Future work includes the fairing and segmentation of curves based on the linearity of these graphs, and most importantly, the extension of logarithmic graphs to surfaces.

References

1. Farin, G.: Curves and Surfaces for CAGD, 5th edn. Morgan Kaufmann Publishers, San Francisco (2002)
2. Farin, G.: Class A Bézier curves. Computer Aided Geometric Design 23(7), 573–581 (2006)

3. Farouki, R.T., Moon, H.P., Ravani, B.: Algorithms for Mikowski products and implicitly-defined complex sets. Advances in Computational Mathematics 13(3), 199–229 (2000)
4. Fukuda, R., Yoshida, N., Saito, T.: Interactive Control of 3D Class-A Bezier Curves, SIGGRAPH ASIA, Sketches (2008)
5. Gray, A., Abbena, E., Salamon, S.: Modern Differential Geometry of Curves and Surfaces with Mathematica, 3rd edn. Chapman & Hall, Boca Raton (2006)
6. Harada, T.: Study of Quantitative Analysis of the Characteristics of a Curve. FORMA 12(1), 55–63 (1997)
7. Harada, T., Yoshimoto, F., Moriyama, M.: An aesthetic curve in the field of industrial Design. In: Proceedings of IEEE Symposium on Visual Languages, pp. 38–47 (1999)
8. Klass, R.: Correction of local surface irregularities using reflection lines. Computer Aided Design 12(2), 73–77 (1980)
9. Miura, K.T.: A general equation of aesthetic curves and its self-affinity. Computer-Aided Design and Applications 3(1-4), 457–464 (2006)
10. Poeschl, T.: Detecting surface irregularities using isophotes. Computer Aided Geometric Design 1(2), 163–168 (1984)
11. Yoshida, N., Saito, T.: Interactive Aesthetic Curve Segments: The Visual Computer (Proc. of Pacific Graphics) 22(9-11), 896–905 (2006)
12. Yoshida, N., Saito, T.: Quasi-Aesthetic Curves in Rational Cubic Bézier Forms. Computer-Aided Design and Applications 4(1-4), 477–486 (2007)
13. Yoshida, N., Hiraiwa, T., Saito, T.: Interactive Control of Planar Class A Bézier Curves using Logarithmic Curvature Graphs. Computer-Aided Design and Applications 5(1-4), 121–130 (2008)

Author Index