Geometry and Computing 10

Tor Dokken
Georg Muntingh
Editors

SAGA – Advances in ShApes, Geometry, and Algebra

Results from the Marie Curie Initial Training Network

Springer

Geometry and Computing

10

More information about this series at
http://www.springer.com/series/7580

Tor Dokken • Georg Muntingh
Editors

SAGA – Advances in ShApes, Geometry, and Algebra

Results from the Marie Curie Initial Training Network

Springer

Editors
Tor Dokken
Georg Muntingh
SINTEF ICT
Oslo
Norway

ISSN 1866-6795 ISSN 1866-6809 (electronic)
ISBN 978-3-319-08634-7 ISBN 978-3-319-08635-4 (eBook)
DOI 10.1007/978-3-319-08635-4
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014953535

Mathematics Subject Classification (2010): 13P25, 14Q10, 65D17, 65D18, 65D07, (13D02), 41A10, 49M99, 65M60, 65N30

Cover illustration: Heidi E. I. Dahl

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

SAGA – ShApes, Geometry and Algebra – was a Marie Curie Initial Training Network (ITN) organized from 2008 to 2012 and intended to offer researchers in the first five years of their careers the opportunity to improve their research skills, join established research teams and enhance their career prospects.

As a common requirement in an ITN, researchers were usually expected to exhibit transnational mobility, i.e., move from one country to another. In addition to crossing geographic borders, SAGA offered mobility between the closely related, but nearly disjoint, scientific communities of Geometric Modeling and Algebraic Geometry. Projects funded by the European Union have played a central part in bridging the gap between these communities, with the first steps taken from 2001 to 2005 in the Fifth Framework Future and Emerging Technologies project GAIA – Intersection Algorithms for Geometry-based IT, which utilized approximate algebraic methods.

For young and seasoned researchers alike, SAGA has offered an inspiring venue for exchanging ideas and forming new connections for cooperation. The young researchers of SAGA have not only established relations with researchers at their own institutions, but also with researchers at other institutions working in their field of interest, through the secondments to other partners and participation in the workshops and events. That being said, the most important aspect for the future careers of these young researchers is most likely the basis for future cooperation and research they have established among themselves.

This book provides insights into research conducted in the SAGA ITN and represents its long-term documentation. It consists of an introduction and 14 chapters, divided into four parts according to the work packages in the SAGA ITN, and written by a combination of young and established researchers.

The editors are grateful to the reviewers for their reports, which made it possible to select chapters suitable for publication and to improve them considerably. In addition, we would like to thank the lead scientists for their continuing devotion to SAGA. Special thanks go to Ewald Quak, for his central role in the organization of events and ensuring quality throughout. We would also like to express our gratitude to all external contributors, among whom Tom Grandine deserves special mention for his excellent talks at the events connecting mathematics to the industry. Finally, we would like to thank Springer, in particular Martin Peters and Ruth Allewelt, for the pleasant and smooth cooperation in preparing this volume. The research leading to these findings was supported by funding from the European Community's Seventh Framework Program under grant agreement no. PITN-GA-2008-214584-SAGA.

Oslo, Norway
May 2014

Tor Dokken
Georg Muntingh

Contents

Part I Change of Representation

Part II Geometric Computing: Algebraic Tools

Chapter 1
Introduction to ShApes, Geometry, and Algebra

Tor Dokken and Georg Muntingh

1.1 Motivation

Computer Aided Design and Manufacturing (CAD/CAM) originated nearly half a century ago and is now a key part of product development and the production process.

During the last decade, technologies for information processing and data acquisition have made rapid progress. The availability of cheap computing power on desktop computers, including high-performance graphics capabilities, and the advent of laser scanners for 3D objects, which are able to digitize even complex geometric models within seconds, pose new challenges for CAD/CAM. The geometric models have become more complex, and the use of this technology has become far more widespread, especially among small and medium-sized enterprises.

The technology represented by the STEP standard (ISO 10303, Automation systems and integration – Product data representation and exchange) is no longer adequate to address the mathematical problems that arise. In the long term, this development may prove to be problematic, since it decouples the CAD/CAM industry from new applications such as mobile telecommunication and the computer game industry. CAD systems are due for major rework to be able to exploit the computational performance of multi-core CPUs and data stream accelerators such as programmable graphics cards. Experiments show that data stream accelerators outperform CPUs by an order of magnitude for tasks that can be effectively parallelized. Current hardware has three to four orders of magnitude more computational performance than the hardware standard CAD technology originally targeted. This presents the opportunity to use more advanced approaches in CAD systems.

T. Dokken • G. Muntingh (✉)
SINTEF ICT, P.O. Box 124 Blindern, N-0314 Oslo, Norway
e-mail: Tor.Dokken@sintef.no; Georg.Muntingh@sintef.no

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_1

Within the numerical analysis community, the use of higher order polynomial representations (hpFEM and the isogeometric approach) has been conceived as a new way to break the complexity barrier caused by piecewise linear representations, and to deal efficiently with free-form geometry. In order to exploit the potential of these developments, this progress has to be matched by corresponding research and development in the geometric and CAD/CAM community.

1.2 The SAGA Initial Training Network

To address the above issues, the European Commission funded ShApes, Geometry and Algebra (SAGA, 2008–2012). As a Marie Curie Actions Initial Training Network (ITN), its main objective was to recruit and train young researchers, i.e., researchers with less than 5 years of research experience, in a multi-national network of academic and industrial research partners. These partners, and their *representatives*, were:

- *Dr. Tor Dokken* (Chief Scientist and SAGA coordinator)
  SINTEF, Department of Applied Mathematics, Oslo, Norway;
- *Prof. Ragni Piene*
  University of Oslo, CMA/Department of Mathematics, Norway;
- *Prof. Bert Jüttler*
  Johannes Kepler Universität, Institute of Applied Geometry, Linz, Austria;
- *Dr. Bernard Mourrain*
  Institut National de Recherche en Informatique et Automatique, INRIA Sophia-Antipolis Research Center, France;
- *Prof. Laureano Gonzalez-Vega*
  Universidad de Cantabria, Departamento de Matematicas, Estadistica y Computacion, Santander, Spain;
- *Prof. Rimvydas Krasauskas*
  Vilniaus universitetas, Faculty of Mathematics and Informatics, Computer-aided Geometry Lab, Lithuania;
- *Prof. Ioannis Emiris*
  National and Kapodistrian University of Athens, Department of Informatics and Telecommunications, Greece;
- *Dr. Raffaele De Amicis* (director of GraphiTech)
  Fondazione GraphiTech, Trento, Italy;
- *Mr. Dominique Laffret* (vice president Strategic Partnerships)
  Missler Software, France;
- *Prof. Eigil Samset*
  Kongsberg SIM, Norway.

SAGA employed 14 Early Stage Researchers (ESRs), i.e., researchers with less than 4 years of research experience at the time of recruitment. ESRs without a PhD degree were expected to pursue one, especially the 7 fellows that were granted a

36 months fellowship. By the summer of 2014, six of these had defended their PhD, and the remaining fellow will defend her PhD in the autumn of 2014. Of the remaining ESRs, three used their time to complete their PhD. SAGA also recruited eight Experienced Researchers (ERs), i.e., Postdoctoral Research Fellows, for a time period of 3–12 months. In addition the network financed 5 Visiting Scientists for 1 or 2 months.

The researchers were recruited from many different countries, with ESRs from China, Colombia, Denmark, France, Germany, Greece, India, Lithuania, Niger, Norway, Portugal, United Kingdom, Vietnam (2×), and ERs from Argentina, Canada, France, Italy (3×), Norway and Vietnam. The Visiting Scientists came from Argentina, Czech Republic, Japan, Spain and the United States. However, the network reached much further, as the SAGA kick-off event, winter and autumn schools attracted many young and established researchers from outside the network as well, whose attendance was partly funded by SAGA. There were five such events [22–24]:

- SAGA kick-off in Castro Urdiales, November 17–21, 2008, Spain,
  47 participants from outside SAGA;
- Auron Winter School, March 15–19, 2010, France,
  30 participants from outside of SAGA;
- Kolympari Autumn School, October 4–8, 2010, Greece,
  39 participants from outside of SAGA;
- Vilnius Autumn School, September 27–30, 2011, Lithuania,
  34 participants from outside of SAGA;
- SAGA Final Conference in Trento, October 9–11, 2012, Italy,
  31 participants from outside of SAGA.

In addition SAGA mini-symposia and special sessions were organized in various major international conferences, to give the SAGA fellows the opportunity to present their results as a group as well, for example at

- The SIAM Conference on Applications of Algebraic Geometry, October 8–12, 2011, in Raleigh, North Carolina, USA;
- The 8th International Conference on Mathematical Methods for Curves and Surfaces, June 28–July 3, 2012, in Oslo, Norway.

SAGA aimed to promote the relatively new field of Approximate Algebraic Geometry, which bridges traditional fields like Computer-Aided Geometric Design, Real Algebraic Geometry, Computer Algebra, Numerical Analysis, and Approximation Theory. In addition, the project had as a goal to strengthen interdisciplinary research and development concerning CAD/CAM, by training a new generation of researchers familiar with both academic and industry viewpoints, while supporting the cooperation among the partners and with other interested collaborators in Europe. We briefly mention some research highlights in SAGA:

- Refinement and development of an easy to compute non-square matrix-based implicit representation for parameterized rational planar curves, space curves and surfaces [6, 7, 17].
- Development of a new algorithm for isolating the real roots of a system of multivariate polynomials, given in the monomial basis [18].
- New combinatorial lower and upper bounds for the dimension of spline spaces over planar T-meshes and triangulated domains [19, 26].
- Exploiting sparsity of the input parametric representation in exact and approximate implicitization with a new method that is oblivious of base points and reduces to a matrix kernel computation, thus leading to fast methods amenable to approximate computation [12–16].
- A novel semi-automatic procedure to model the geometry of wooden elements [25].
- Improved blends between primitive CAD surfaces, and new results on exact rational parameterizations for fixed and variable radius rolling ball blends of pairs of natural quadrics [8, 9].
- An initial understanding was established for the structure of the spline ring as a generalized Stanley-Reisner ring [26]. In toric geometry, several new results concerning the characterization of lattice polytopes related to higher discriminants were obtained [10, 11].
- Hausdorff limits of toric patches and degenerations were studied, resulting in a new and elementary interpretation of the secondary polytope [21].
- A novel approach to build a bijective parameterization for a contractible domain in $\mathbb{R}^3$ with a significant potential for use in isogeometric analysis [20].
- Approximate implicitization has been extended from only using the Bernstein basis to the use of Jacobi polynomials such as Chebyshev polynomials. The use of Jacobi polynomials allows for assembling the matrix of approximate implicitization by using algorithms similar to Fast Fourier Transform that significantly improve the assembly process. The use of Chebyshev polynomials significantly increases the accuracy of approximate implicitization [1–5].

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement no. PITN-GA-2008-214584-SAGA.

1.3 A Preview of This Book

The remainder of this volume is a collection of research results obtained in the SAGA ITN, written by a combination of young researchers and lead scientists. It is divided into four parts, each corresponding to a work package in the project. We proceed with a brief summary of each chapter.

Part I: Change of Representation

Barrowclough presents an accessible discussion of several approaches to computing implicitizations numerically, including both exact methods for low degree curves and approximate methods for higher degree surfaces and envelopes.

Emiris, Kalinka, and Konaxis describe an implicitization algorithm based on predicting the support of the implicit equation from specialized resultants, whose implementation is efficient for common instances in geometric modeling.

Luu Ba studies the conversion of parametric curves and surfaces into a matrix-based implicit representation, in which the ambient space is stratified into points with multiplicity equal to the corank of the matrix, i.e., points outside the variety, regular points on the variety, singular points of multiplicity 2 on the variety, etc.

Part II: Geometric Computing: Algebraic Tools

Mantzaris and Mourrain review recent advances for handling isolated singularities of polynomial ideals, based on computing a basis for the dual space and of the local quotient ring at a given (approximate) singular point, deflating the system by augmenting it with new equations derived from the dual basis, and certifying the singular point and its multiplicity structure for the perturbed system.

Dickenstein, Emiris and Karasoulou present an original formula that relates the (mixed sparse) discriminant of two bivariate (Laurent) polynomials to the sparse resultant of these polynomials and their toric Jacobian, which yields as a corollary the known formula for the bidegree of the discriminant and a novel multiplication formula for the discriminant when one of the polynomials factors.

Chau and Galligo provide a subdivision algorithm for computing an approximation of the intersection curve between two parametric surfaces, represented in 4D parameter space, for which the exactness of the topology is certified up to any specified precision.

Krasauskas and Zubė provide simple, elegant extensions of results about quaternion Bézier curves and surfaces to analogous results about Bézier curves and surfaces in various Clifford algebras, in order to develop an intuitive approach to defining and manipulating such Bézier curves and surfaces.

Part III: Algebraic Geometry for CAD Applications

Piene presents recent developments in “algebraic spline geometry”, and provides two examples that support the Local Spline Ring Conjecture on the geometry of generalized Stanley-Reisner rings, which are the rings $C^r(\Delta)$ of r-smooth piecewise polynomials on a simplicial complex $\Delta \subset \mathbb{R}^d$. Roughly, the statement is that the affine scheme corresponding to the generalized Stanley-Reisner ring $C^r(\Delta)$ is a “twisted” version of a topological embedding of Δ, in which the irreducible components intersect with an intersection multiplicity corresponding to the smoothness r.

Villamizar and Mourrain derive lower and upper bounds for the dimension of the space of r-smooth piecewise polynomials of degree d defined on triangular and tetrahedral partitions, using the homological approach initiated by Billera. The formulas take into account the geometry of the faces surrounding the interior faces of the partition and make use of Fröberg’s conjecture on the dimension of ideals generated by a generic set of forms in a polynomial ring.

Postinghel presents recent results on a class of polynomial interpolation problems that amount to determining the dimension of linear systems of homogeneous or multi-homogeneous polynomials vanishing together with their partial derivatives at a finite set of general points, employing algebro-geometric techniques such as blowing-up and degenerations. These results include a proof of a special case of Fröberg’s conjecture, a formula for the dimensions of linear systems with general points of any multiplicity in $\mathbb{P}^n$ in a family of cases for which the base locus is only linear, and a complete classification of the linear systems with double points in general position in products $(\mathbb{P}^1)^n$ of projective lines.

Dahl presents a new method for constructing rational variable radius rolling ball blends of natural quadrics, based on placing suitable control spheres at strategic points. The method is illustrated by a detailed example of a composite configuration with multiple edges and 3-sided corners.

Part IV: Practical Industrial Problems

Adamou, Fioravanti, Gonzalez-Vega, and Mourrain present new approaches for computing an exact algebraic parametrization of a planar curve-curve bisector and two low degree rational surfaces bisector, and an automatic geometric and numerical characterization of the planar point-curve and curve-curve bisector. In addition, they describe a new algorithm for computing a Voronoi diagram of a set of parallel half-lines in three-dimensional space, constrained to a compact domain.

Nguyen presents an algorithm for constructing a bijective parametrization of a contractible domain in $\mathbb{R}^3$ from a triangulation of the boundary using trivariate tensor-product B-splines. This mapping is defined via a sequence of harmonic maps and modified by a reparameterization to agree with the boundary parameterization, and the final spline representation for the domain is constructed by approximating the inverse of the computed mapping.

Nørtoft and Dokken extend to Locally Refinable B-splines two well-known tensor-product flow discretizations, namely the Taylor-Hood and the multigrid discretizations, and apply these to solve the mixed formulation of the steady-state, incompressible Navier-Stokes equations in two dimensions using isogeometric analysis. The chapter ends with a series of numerical investigations of the use of

Locally Refinable B-splines in isogeometric analysis of flow problems, including the stability of the discretizations, error convergence during refinement based on a manufactured solution, and benchmarking based on the lid-driven cavity problem.

References

1. O.J.D. Barrowclough, Approximate methods for change of representation and their applications in CAGD. Ph.D. thesis, University of Oslo, 2013
2. O.J.D. Barrowclough, A basis for the implicit representation of planar rational cubic Bézier curves. Comput. Aided Geom. Des. **31**(3–4), 148–167 (2014). doi:http://dx.doi.org/10.1016/j.cagd.2014.02.007. http://www.sciencedirect.com/science/article/pii/S0167839614000223
3. O.J.D. Barrowclough, T. Dokken, Approximate implicitization of triangular Bézier surfaces, in *Proceedings of the 26th Spring Conference on Computer Graphics, SCCG'10*, Budmerice Castle, Solvakia (ACM, New York, 2010), pp. 133–140 doi:10.1145/1925059.1925084
4. O.J.D. Barrowclough, T. Dokken, Approximate implicitization using linear algebra. J. Appl. Math. (2012). doi:10.1155/2012/293746
5. O.J.D. Barrowclough, B. Jüttler, T. Schulz, Fast approximate implicitization of envelope curves using Chebyshev polynomials, in *Latest Advances in Robot Kinematics*, ed. by J. Lenarčič, M. Husty (Springer, Dordrecht, 2012), pp. 205–212. doi:10.1007/978-94-007-4620-6_26
6. L. Busé, T. Luu Ba, Matrix-based implicit representations of rational algebraic curves and applications. Comput. Aided Geom. Des. **27**(9), 681–699 (2010). doi:10.1016/j.cagd.2010.09.006. http://dx.doi.org/10.1016/j.cagd.2010.09.006
7. L. Busé, T. Luu Ba, The surface/surface intersection problem by means of matrix based representations. Comput. Aided Geom. Des. **29**(8), 579–598 (2012). doi:10.1016/j.cagd.2012.04.002. http://dx.doi.org/10.1016/j.cagd.2012.04.002
8. H.E.I. Dahl, Piecewise rational parametrizations of canal surfaces, in *Mathematical Methods for Curves and Surfaces*, eds. by M. Floater, T. Lyche, M.L. Mazure, K. Mørken, L.L. Schumaker, Lecture Notes in Computer Science, vol. 8177 (Springer Berlin Heidelberg, 2014), pp 88–111. doi:10.1007/978-3-642-54382-1_6. http://dx.doi.org/10.1007/978-3-642-54382-1_6
9. H.E.I. Dahl, R. Krasauskas, Rational fix radius rolling ball blends between natural quadrics. Comput. Aided Geom. Des. **29**, 691–706 (2012)
10. A. Dickenstein, S. Di Rocco, R. Piene, Higher order duality and toric embeddings. Annales de l'Institut Fourier (2014, to appear). http://arxiv.org/abs/1111.4641
11. A. Dickenstein, R. Piene, Higher order selfdual toric varieties (in progress)
12. I. Emiris, C. Konaxis, L. Palios, Computing the Newton polygon of the implicit equation. Math. Comput. Sci. Spec. Issue Comput. Geom. Comput. Aided Des. **4**(1), 25–44 (2010)
13. I. Emiris, I. Kotsireas, Implicit polynomial support optimized for sparseness, in *Proceedings of the International Conference on Computational Science and Applications: Part III*, Montreal (Springer, Berlin, 2003), pp. 397–406. http://portal.acm.org/citation.cfm?id=1761792.1761839
14. I.Z. Emiris, V. Fisikopoulos, C. Konaxis, L. Peñaranda, An output-sensitive algorithm for computing projections of resultant polytopes, in *Proceedings of the 28th Annual Symposium on Computational Geometry, SoCG'12*, Chapel Hill (ACM, North Carolina, USA, 2012), pp. 179–188. doi:10.1145/2261250.2261276. http://doi.acm.org/10.1145/2261250.2261276.
15. I.Z. Emiris, T. Kalinka, C. Konaxis, T. Luu Ba, Implicitization of curves and (hyper)surfaces using predicted support. Theor. Comput. Sci. Spec. Issue **479**(0), 81–98 (2013). doi:10.1016/j.tcs.2012.10.018. http://www.sciencedirect.com/science/article/pii/S0304397512009395

16. I.Z. Emiris, T. Kalinka, C. Konaxis, T. Luu Ba, Sparse implicitization by interpolation: characterizing non-exactness and an application to computing discriminants. Comput. Aided Des. Spec. Issue Symp. Phys. Model. **45**(2), 252–261 (2013)
17. T. Luu Ba, L. Busé, B. Mourrain, Curve/surface intersection problem by means of matrix representation, in *SNC'09: Proceedings of the International Conference on Symbolic Numeric Computation*, Kyoto (ACM, New York, 2009), pp. 71–78
18. A. Mantzaflaris, B. Mourrain, Deflation and certified isolation of singular zeros of polynomial systems, in *ISSAC 2011—Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*, San Jose (ACM, New York, 2011), pp. 249–256. doi:10.1145/1993886.1993925. http://dx.doi.org/10.1145/1993886.1993925
19. B. Mourrain, N. Villamizar, Homological techniques for the analysis of the dimension of triangular spline spaces. J. Symb. Comput. **50**, 564–577 (2013). doi:10.1016/j.jsc.2012.10.002. http://dx.doi.org/10.1016/j.jsc.2012.10.002
20. T. Nguyen, B. Jüttler, Parameterization of contractible domains using sequences of harmonic maps, in *Curves and Surfaces*. Lecture notes in compututer science, vol. 6920 (Springer, Heidelberg, 2012), pp. 501–514. doi:10.1007/978-3-642-27413-8_32. http://dx.doi.org/10.1007/978-3-642-27413-8_32
21. E. Postinghel, F. Sottile, N. Villamizar, Degenerations of irrational toric varieties (2013, preprint). http://arxiv.org/abs/1312.3834
22. SAGA, *Kick-Off Event*, Castro Urdiales, 17–21 Nov 2008. http://www.saga_ciem.unican.es/
23. SAGA, *Fall School on Shapes, Geometry, and Algebra*, Kolympari, 4–8 Oct 2010. http://erga.di.uoa.gr/sagaschool/
24. SAGA, *3rd Saga Workshop*, Trento, 9 Oct 2012. http://www.graphitech.it/SAGA2012/
25. B. Simōes, M. Riggio, R. De Amicis, Modeling morphological features of timber from x-ray tomographic images. Int. J. Interact. Des. Manuf. (IJIDeM) **6**(2), 65–73 (2012). doi:10.1007/s12008-012-0140-5. http://dx.doi.org/10.1007/s12008-012-0140-5
26. N. Villamizar, Algebraic geometry for splines. Ph.D. thesis, University of Oslo, 2012. http://urn.nb.no/URN:NBN:no-38634

Part I
Change of Representation

Chapter 2
Numerical Methods for Implicitisation and Their Applications

Oliver J.D. Barrowclough

2.1 Introduction

Mathematical representation of geometric objects plays an important role in fields such as automotive and aeronautical engineering, architecture, computer graphics, animation and robotics. Modern CAD systems are based on representations that encompass a very general set of freeform curves and surfaces. The two main representations of such objects are the *parametric* and *implicit* forms. Of these, the parametric form is by far the most prevalent. This is mainly due to the ease of generating points lying on parametric curves and surfaces. The natural geometric properties of Bézier and B-spline parametric forms have also supported their dominance. However, implicit forms have several useful properties that complement those of the parametric representation. For example, deciding whether a given point lies inside, outside or exactly on a given curve or surface, is simplified if an implicit representation is available. Implicit representations also naturally support operations such as intersections, unions, differences and offsets. For these reasons, methods for *change of representation* are of great practical importance.

Many of the curves and surfaces used in CAD have very simple forms that also exhibit simple representations. In 2D, these include lines, ellipses, parabolas and hyperbolas, all of which come under the common term of conic sections. In 3D, such surfaces include planes, spheres, cylinders of algebraic degree 2, as well as tori, which are of degree 4. In both the 2D and 3D cases, these shapes can be represented both parametrically and implicitly. However, more advanced models, such as those seen in automobile design, aircraft wings, ship hulls and wind turbines require more general freeform representations.

O.J.D. Barrowclough (✉)
SINTEF ICT, P.O. Box 124 Blindern, N-0314 Oslo, Norway
e-mail: Oliver.Barrowclough@sintef.no

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_2

It is well known that any rational parametric curve or surface can be written in implicit form. The conversion process is known as *implicitisation*. Since parametric forms have become the prevalent representation at the design stage of an application, it is the problem of implicitisation that has received most attention from the research community. Moreover, the conversion in the opposite direction, known as *parametrisation*, is not always possible, since implicit representations encompass a larger class of shapes.[1] In this chapter we discuss several methods for implicitisation including exact methods for low degree curves, and approximate methods for higher degree curves and surfaces. We also look briefly at the implicitisation of envelope curves and surfaces. During the discussion we highlight some of the challenges that are faced in constructing computationally attractive implicitisation algorithms. The work is motivated both as a continuation of recent research into approximate algebraic geometry, and a desire to construct methods better suited for numerical computation on modern architectures. The ever increasing level of parallelism in commodity computers, both with multicore CPUs and GPUs has removed the barrier for the implementation of some computationally intensive methods. Moreover, high quality real-time visualisation methods for implicit surfaces are now available on modern hardware. It is thus important for computationally efficient methods of generating implicitly represented geometries to be available.

An in-depth review of exact implicitisation methods using symbolic computation can be found in [24]. In contrast, this chapter highlights methods that are suitable for numerical computation, although most of the methods can be implemented symbolically if exact precision is required. Methods that support implementation in floating point arithmetic are important for reasons of performance. Moreover, floating point arithmetic is required for many of the linear algebra based approximation algorithms utilised in the methods presented here. Symbolic and infinite precision methods will provide exact results, but this comes at the expense of severely slower performance in almost all cases. Since this chapter is motivated by applications for implicitisation, we restrict our attention to real affine space as opposed to the more general complex projective space. Prototype implementations of the algorithms have been tested using several programming languages including Python, C++ and CUDA.

Several of the papers that we cite herein were written as part of the EU research project SAGA. Among the aims of SAGA was to conduct further research into methods for change of representation and several new results in the field of implicitisation have emerged as a result of the project. This chapter summarises some of these results.

This chapter is structured as follows: We begin, in Sect. 2.2, with a description of specific techniques for implicitisation of low degree rational planar curves. In Sect. 2.3 we present methods, both exact and approximate, for implicitisation of rational parametric curves, surfaces and hypersurfaces. In Sect. 2.4 we discuss

[1] Note that we restrict our attention to *rational parametrisations* in this paper. Other parametrisations of non-rational algebraic curves are possible.

briefly how the methods can be extended to implicitisation of envelope curves and surfaces. We conclude, in Sect. 2.5, with some examples of how implicitisation algorithms can be used in applications.

2.2 Implicitisation Methods for Low Degree Planar Rational Curves

In this section we begin from the very simplest planar curves, describing representations both in the parametric and implicit form. The common parametric representation we will use is the rational Bézier representation. This is defined for a sequence of control points $(\mathbf{c}_0, \ldots, \mathbf{c}_n)$, with $\mathbf{c}_i \in \mathbb{R}^2$, and a sequence of associated weights $(w_0, \ldots, w_n)$, with $w_i \in \mathbb{R}$ as

$$\mathbf{p}(t) = \frac{\sum_{i=0}^{n} w_i \mathbf{c}_i \binom{n}{i} t^i (1-t)^{n-i}}{\sum_{i=0}^{n} w_i \binom{n}{i} t^i (1-t)^{n-i}}, \tag{2.1}$$

for $t \in \boldsymbol{\Omega} \subseteq \mathbb{R}$. The region of interest of the curve is defined by the points corresponding to the parameters t in the domain $\boldsymbol{\Omega}$, which is often the unit interval $[0, 1]$. Any rational curve in the affine plane can be parametrised in the rational Bézier form.

Several exact methods for general degree rational planar curve implicitisation exist; see, for example [8, 10, 21, 34]. In this section we look at alternative methods that define the implicit polynomial explicitly in terms of the control points and weights. This has the advantage of giving some geometric intuition to the construction.

2.2.1 Lines in the Plane

The simplest non-trivial curves in the plane are lines. Typically, from a designer's perspective, a line in the plane will be defined by two distinct points $\mathbf{c}_0, \mathbf{c}_1 \in \mathbb{R}^2$. The rational Bézier representation of a line between these points can be given as

$$\mathbf{p}(t) = \frac{w_0 \mathbf{c}_0 (1-t) + w_1 \mathbf{c}_1 t}{w_0 (1-t) + w_1 t}. \tag{2.2}$$

Since the choice of weights w_0, w_1 has no consequence for the design in this case, they are typically set equal to one. Since the Bernstein basis is a partition of unity, the denominator then becomes equal to one.

An irreducible implicit representation of a line in the plane is unique up to a scalar factor. In a Cartesian coordinate system, we can define the implicit polynomial representing the parametric line (2.2) as

$$L_{ij}(x,y) = \begin{vmatrix} x & y & 1 \\ c_{i,0} & c_{i,1} & 1 \\ c_{j,0} & c_{j,1} & 1 \end{vmatrix}. \tag{2.3}$$

By this definition, the norm of the gradient $\|\nabla L_{ij}\|_2$ is equal to the Euclidean distance between the two points $\mathbf{c}_i$ and $\mathbf{c}_j$:

$$\|\nabla L_{ij}\|_2 = \|\mathbf{c}_i - \mathbf{c}_j\|_2. \tag{2.4}$$

In general, we may wish to define such linear forms without reference to a specific coordinate system. It is natural to define the linear form by enforcing condition (2.4)—that the magnitude of the gradient of the linear form containing the two defining points is equal to the Euclidean distance between those points. Making this assumption simplifies the higher degree constructions in the following sections, and is naturally supported by the Cartesian definition (2.3).

2.2.2 Rational Quadratic Bézier Curves in the Plane

Rational quadratic planar curves encompass the entire set of conic sections, including parabolas, hyperbolas and ellipses. In Bézier form they are defined by three control points, and associated weights; that is, with $n = 2$ in (2.1). The three control points also conveniently define a triangle, on which we can define a barycentric coordinate system. By making a judicious choice of basis on this coordinate system, simple expressions for the implicit polynomial coefficients can be found.

The implicit representation of *any* conic section that is defined in Bézier form, with non-collinear control points, is given by the following equation:

$$4w_1^2\tau_0\tau_2 - w_0w_2\tau_1^2 = 0,$$

where τ_0, τ_1 and τ_2 denote the barycentric coordinates defined by the triangle with vertices $\mathbf{c}_0, \mathbf{c}_1$ and $\mathbf{c}_2$ [19]. The equation can also be expressed independently of the coordinate system, in terms of products of linear forms between the points. In that case we have[2]

$$q = 4w_1^2 L_{01}L_{12} - w_0w_2L_{02}^2 = 0,$$

where L_{ij} is defined, as in Sect. 2.2.1, as a linear form whose gradient norm is proportional to the distance between the points $\mathbf{c}_i$ and $\mathbf{c}_j$. The functions $L_{01}L_{12}$

[2] A proof of this can be seen by expanding the determinant in the implicitisation algorithm given in [35].

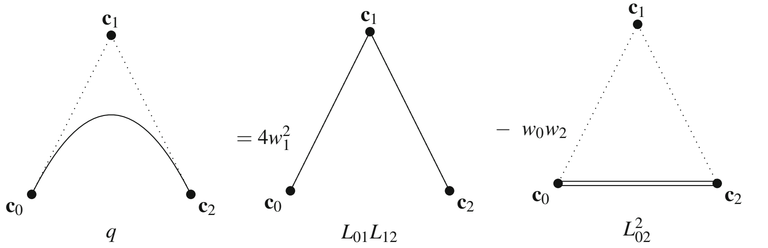


Fig. 2.1 The construction of an implicit representation of rational quadratic Bézier curves in the plane. Here, L_{ij} denotes a linear form containing both $\mathbf{c}_i$ and $\mathbf{c}_j$

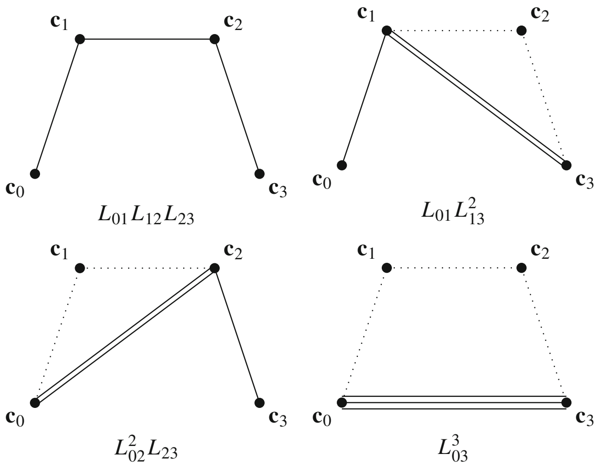
and L_{02}^2 depend on the control points, but provide a basis for all different weight configurations of a given curve, so we refer to them loosely as 'basis functions'.

Figure 2.1 shows a diagrammatic representation of the zero sets of the basis functions used for the implicit representation of rational quadratic Bézier curves. The number of lines between any two points $\mathbf{c}_i$ and $\mathbf{c}_j$, reflect the multiplicity with which L_{ij} appears in the basis function.

2.2.3 Rational Cubic Bézier Curves in the Plane

How best to extend the quadratic method above to cubic and higher degree curves is an interesting problem. For rational cubic curves, one approach, described by Floater in [20], is to choose a Bernstein basis over the triangle defined by the points $\mathbf{c}_0$, $\mathbf{c}_3$ and the intersection point $\mathbf{d}$, of the lines L_{01} and L_{23}. In terms of this basis, the coefficients of the implicit polynomial can be given by explicit formulas. In particular, four of the ten coefficients are immediately zero; thus, some sparsity in the implicit polynomial representation is exhibited. Floater's method suffers from problems when the tangent lines to the curve at the end points $\mathbf{c}_0$ and $\mathbf{c}_3$ are parallel, since in this case the point $\mathbf{d}$ is undefined. Also, if either of $\mathbf{c}_1$ or $\mathbf{c}_2$ lie on the line between $\mathbf{c}_0$ and $\mathbf{c}_3$, the coordinate system collapses.

A new approach to computing the implicit representation of rational planar cubic Bézier curves is presented in [4]. This method requires only four basis functions, which are pictured in Fig. 2.2, and fails only in the case of collinear control points. The method also exhibits simple expressions for the coefficients, and moreover, a great deal of information about the geometry of the curve can be extracted from those coefficients. For example, very simple methods for identifying when unwanted self-intersections occur are presented, along with classifications of the type of curve (i.e., either cuspidal, self-intersecting or exhibiting an acnode). In addition, an explicit formula for the position of the double point is given in terms of a barycentric combination of the control points. We refer the reader to [4] for a fuller description



$L_{01}L_{12}L_{23}$ $L_{01}L_{13}^2$

$L_{02}^2L_{23}$ L_{03}^3

Fig. 2.2 A diagrammatic representation of the zero sets of the basis functions for rational planar cubic curves

of the method and its consequences. Another approach with some similar results is described by Stone and DeRose in [38].

2.2.4 *Rational Bézier Curves of Higher Degree*

Explicit methods for implicitisation become more complex as the degree of the curve increases. In the introduction of [3], a set of functions that appear to support all rational quartic planar Bézier curves without collinear control points is presented. These functions are pictured in Fig. 2.3. We make a formal conjecture that this is the case here.

Conjecture 2.2.1 *Suppose we are given a rational quartic Bézier curve in the plane (i.e., with $n = 4$ in (2.1)), such that no three of the control points $\mathbf{c}_0, \dots, \mathbf{c}_4$ are collinear. Then the implicit representation can be written as a linear combination of the following set of nine functions:*

$$\{L_{01}L_{12}L_{23}L_{34},\ L_{01}L_{12}L_{24}^2,\ L_{02}^2L_{23}L_{34},\ L_{01}L_{13}^2L_{34},$$
$$L_{02}L_{04}^2L_{24},\ L_{01}L_{14}^3,\ L_{03}^3L_{34},\ L_{02}^2L_{24}^2,\ L_{04}^4\}.$$

A formal proof of this conjecture is a subject for further research. To what extent the interesting properties seen for cubic curves can be replicated for quartic and higher degrees, also remains to be seen. However, the number of basis functions required to represent the polynomials increases exponentially with degree. In fact, the number of functions required for a curve of degree n, is related to the number of terms in the discriminant of a univariate polynomial of degree n. For higher degrees, this quickly becomes greater than the dimension of the polynomial space (which is

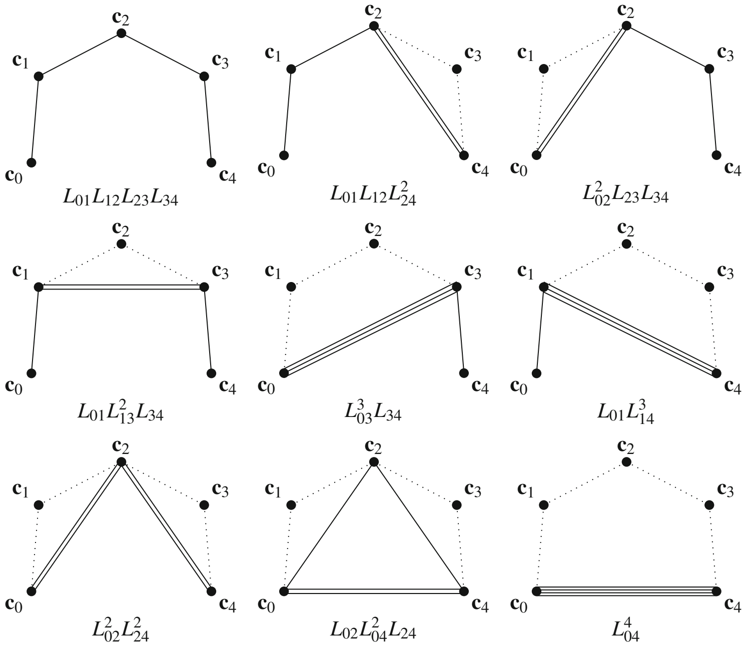


Fig. 2.3 A diagrammatic representation of the zero sets of nine basis functions that appear to support the implicit representation of rational quartic Bézier curves, when no three control points are collinear

$O(n^2)$), leading to linear dependencies between the functions and thus the loss of the basis property. Due to this high level of complexity, it is therefore reasonable to suspect that the methods for higher degrees will be difficult to analyse. However, the potential geometric insight that may be possible with such techniques warrants further investigation.

An often cited drawback of implicit curves and surfaces is that it is difficult to directly control the geometry by manipulating the coefficients of the defining polynomial. However, the methods presented in this section show that elegant geometric structures are also present using implicit representations. Indeed, these methods can be used directly for design using implicit visualisation methods, without reference to the parametric representation at all; though, in practice, the availability of both representations is optimal for design.

Although existing methods for exact implicitisation can be suitable for moderately high degree planar curves, problems begin to arise for surfaces and hypersurfaces. Moreover, it is often infeasible to compute exact implicitisations in floating point arithmetic, due to the limited precision available. Some of the issues with exact implicitisation are highlighted in the following section.

2.3 Numerical Methods for Implicitisation of Higher Degree Curves, Surfaces and Hypersurfaces

For the description of rational parametric surfaces and hypersurfaces, we can also use the rational Bézier representation. This is defined in a similar way to curves, and can be extended by using either tensor-product or simplex structures. For a general approach, we will assume that we have a rational parametric description $\mathbf{p}(\mathbf{t}), \mathbf{t} \in \boldsymbol{\Omega} \subset \mathbb{R}^{d-1}$ of a hypersurface in $\mathbb{R}^d$. Any rational hypersurface in $\mathbb{R}^d$ can be parametrised in the rational Bézier form.

For planar rational curves, the degree of the exact implicit polynomial representation is equal to that of the parametric representation. However, when it comes to surfaces, this is no longer true. For rational surfaces of total degree n defined on a triangular domain, the degree of the implicit polynomial is n^2. Further, for tensor-product parametric surfaces of bi-degree (n_1, n_2), the degree of the implicit polynomial is $2n_1n_2$. For higher dimensional hypersurfaces, the degrees increase even more rapidly.

Traditional methods for implicitisation of surfaces using the theory of elimination and resultants have been known as far back as the nineteenth Century [33]. However, several obstacles have hindered the practical use of these algorithms. Some of the problems, highlighted in the papers [14, 22, 34] among others, include the following:

- **Additional solutions.** It is often the case that the implicit polynomial defined by resultant computation contains factors that are not part of the implicit equation of the curve. Thus, polynomial factorisation is required to find the irreducible polynomial of interest. Factorisation of polynomials is an undesirable operation in CAGD, especially when using floating point coefficients because small perturbations in the coefficients of a reducible polynomial, which occur when approximating with floating point numbers, can render it irreducible.
- **Computationally expensive algorithms.** Computation of the determinant of a matrix with symbolic entries entails $O(n!)$ operations, where n is the dimension of the matrix. Thus, the methods are highly dependent on the size of the matrix, and quickly become slow for high degrees. Often the evaluation times will be unreasonable for applications.
- **Numerical stability.** For reasons of performance, it is often desirable to use floating point numerics. In CAGD, rational parametric curves and surfaces are mostly given in Bernstein form, whereas the traditional techniques for implicitisation typically use the monomial basis. The basis transformations between Bernstein and monomial bases are ill conditioned, and a loss of accuracy is to be expected.
- **High polynomial degrees.** As described previously, the exact algebraic degrees of surfaces can be undesirably high. Sometimes (e.g., if symmetries occur), the algebraic degree will reduce, but extra factors in the implicit equation will occur. High degrees also lead to issues with performance and numerical stability.

- **Base points.** A base point is defined as a point where the numerator and denominator of the parametric description disappear simultaneously. In the case that base points occur, traditional methods for resultant computation will fail (the implicit polynomial will be identically zero). Moreover, base points are a common occurrence even in freeform surfaces [34].
- **Unwanted branches and self-intersections.** Whereas parametric curves and surfaces are defined in a specific parameter domain, implicit curves require a 2D region of interest to be defined. It is quite possible, and often the case, that extra branches or self-intersections occur that are not present in the parametric definition. This is, however, a feature of exact implicitisation, rather than a problem with any specific method. Further techniques such as approximation are required in order to remove them.

Some of these challenges have been resolved in the CAGD literature of the past few decades. For example, Sederberg and Chen introduced the method of moving curves and surfaces which removed the problem of base points, and in fact reduced the complexity in such situations, by reducing the algebraic degree [34]. Other methods, such as those using linear algebra, introduced the ability to approximate parametric surfaces with implicit surfaces. In this section we focus mainly on methods for implicitisation (and approximate implicitisation) that utilise linear algebra. Some approaches to tensor-product implicitization are described in [6, 13], whereas simplex approaches (specialised to surfaces) are described in [5]. The general formulation is quite simple, but it can be specialised in many different ways, each of which exhibit different properties. There also exist techniques that have been developed to improve the performance of the method in certain cases, by exploiting sparsity (e.g., [17, 18]).

2.3.1 A General Method for Implicitisation and Approximation Using Linear Algebra

All of the implicitisation techniques described in this section require a choice of basis. Typical examples include the traditional power or monomial basis, which is the main basis in the theoretical field of algebraic geometry. More applied methods tend to use more numerically stable bases, such as the Bernstein basis. These are generally defined in a barycentric coordinate system which is local to the region of interest of the curve or surface.

The types of basis function that can be used are conceptually different. We give the name *dynamic basis functions*, to bases that change with changes in the geometry of the curve or surface (e.g., local Bernstein bases). This contrasts with static basis functions, such as the monomial basis, which do not depend on the curve or surface in question. Static basis functions can exhibit poor numerical stability if the hypersurface is far from the origin. Well chosen dynamic basis functions have better numerical properties and can reduce the complexity of the computation. For example, the dynamic basis functions defined previously in Sects. 2.2.2–2.2.4 are a good choice for low degree planar curves.[3]

In terms of a general function set $\{q_0, \ldots, q_L : q_i : \mathbb{R}^d \rightarrow \mathbb{R}, i = 0, \ldots, L\}$ we write

$$q = \sum_{i=0}^{L} b_i q_i,$$

for scalar coefficients $\mathbf{b} = (b_i)_{i=0}^{L}$. The task is then to generate the coefficients $\mathbf{b}$, which define the function q, such that $q \circ \mathbf{p} \equiv 0$, where $\mathbf{p}$ is the parametric description defined at the beginning of Sect. 2.3. Often it is not possible, or indeed desirable, to achieve this exactly. In such cases, approximations of q, such that $q \circ \mathbf{p} \approx 0$ within the domain $\boldsymbol{\Omega}$ may be preferable. This is done by attempting to minimise the algebraic error $|q \circ \mathbf{p}|$. Minimising the error in affine space is a computationally intractable problem. However, the algebraic error provides a good approximation to the geometric error away from singularities [13].

By substituting the parametric description into the (unknown) implicit form, we obtain

$$q \circ \mathbf{p}(\mathbf{t}) = \boldsymbol{\alpha}(\mathbf{t})^T \mathbf{D}\mathbf{b},$$

where $\mathbf{D}$ is a matrix of coefficients and $\boldsymbol{\alpha}(\mathbf{t})$ is a vector of basis functions of the function $q \circ \mathbf{p}$ in the parameter $\mathbf{t}$. In the exact case, the null space of $\mathbf{D}$ will give the desired coefficients. In the approximate case, taking the right singular vector corresponding to the smallest singular value in a singular value decomposition (SVD), will generally give good approximations. A more detailed description of different approaches to approximate implicitisation using linear algebra is given in [6]. A short summary of some techniques is provided in the following sections.

2.3.2 Approximate Implicitisation

The original method for approximate implicitisation was introduced in 1997 in the doctoral thesis of T. Dokken [13, 14]. This approach allows the degree of the implicit polynomial to be chosen, and the algorithm proceeds to find an algebraic approximation. The method, which utilises the beneficial properties of Bézier and Bernstein representations, was extended to several other polynomial bases in [6].

[3] Note that it is not necessary that the function set forms a basis for the entire set of multivariate polynomials of a certain degree in $\mathbb{R}^d$, but it is necessary that the function set supports the implicit representation.

All of these methods exhibit the same very high convergence rates, justifying their suitability for approximate approaches to implicitisation. These methods are designed to provide approximations, but also have the potential to provide exact implicitisations if the chosen degree is high enough, and exact precision arithmetic is used.

Dokken's original approach, using partition of unity bases, finds a polynomial q that provides an upper bound on the maximum values of the algebraic error in terms of the smallest singular value $\sigma_{\min}$ of the matrix $\mathbf{D}$:

$$|q \circ \mathbf{p}(\mathbf{t})| \leq \sigma_{\min}\,, \quad \text{for all } \mathbf{t} \in \boldsymbol{\Omega}\,.$$

An alternative approach, known as weak approximate implicitisation, which minimises the least squares error was introduced later [12, 15]:

$$\min_{\|\mathbf{b}\|_2=1} \int_{\boldsymbol{\Omega}} (q \circ \mathbf{p}(\mathbf{t}))^2 \,\mathrm{d}\mathbf{t}. \tag{2.5}$$

This approach also uses SVD approximation, but is different from the approach outlined in Sect. 2.3.1 in that the matrix coefficients are generated by performing integrals. This method is very general, and tends to give better approximations than Dokken's original approach.

It was shown in [6] that by using orthonormal polynomial bases, the same theoretical results as for weighted least squares problems can be achieved, but with much better numerical stability than the weak approach. In particular, when the parametric hypersurface is described in tensor-product form, approximate implicitisation in the Chebyshev basis gives several computational and approximative advantages. For example, the Chebyshev coefficients can be generated efficiently using Fast Fourier Transforms. Additionally, the maximum error on the domain of interest was shown experimentally to be far superior to that of the Bernstein basis. The Bernstein basis did, however, outperform all other bases with respect to numerical stability, when implicitising with the correct degree.

Figure 2.4 shows how the values are distributed in the matrices for four different bases. Note that the distribution of singular values is much more even in the Bernstein basis than the other bases. For the Chebyshev basis, the values lower down in the matrix degenerate to the extent that the bottom half of the matrix hardly contributes to the approximation. This is potentially useful for large problems, where large areas of the matrix could be discarded prior to SVD in order to improve performance.

Alternative approaches to approximate implicitisation have also been introduced. Wurm and Jüttler introduced a method in [23, 40] that approximates scattered data by implicit surfaces. Shen et al. introduced the concept of approximate μ-bases, which led to a different approach to approximate implicitisation [36]. Their work also has applications to the degree reduction of curves.

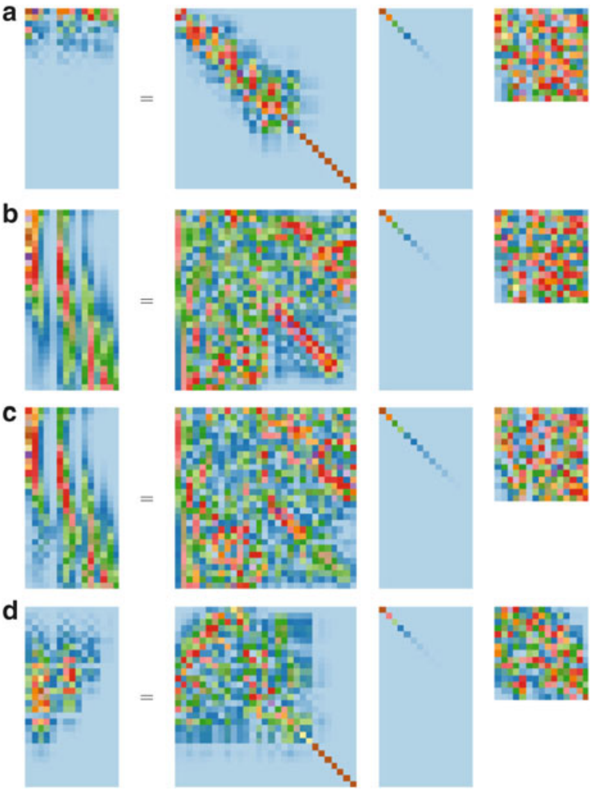


Fig. 2.4 Implicitisation matrices (*left*) and their singular value decompositions, highlighting the structures for the various polynomial bases. The *light blue* elements are zero or close to zero. (**a**) $\mathbf{D}_C = \mathbf{U}_C \boldsymbol{\Sigma}_C \mathbf{V}_C^T$ for the Chebyshev basis. (**b**) $\mathbf{D}_L = \mathbf{U}_L \boldsymbol{\Sigma}_L \mathbf{V}_L^T$ for the Lagrange basis. (**c**) $\mathbf{D}_B = \mathbf{U}_B \boldsymbol{\Sigma}_B \mathbf{V}_B^T$ for the Bernstein basis. (**d**) $\mathbf{D}_M = \mathbf{U}_M \boldsymbol{\Sigma}_M \mathbf{V}_M^T$ for the monomial basis

2.3.3 Interpolation and Approximation of Point Data

One of the simplest and potentially most powerful approaches to approximate implicitisation is using point data for interpolation or approximation. The theory of interpolation with implicit hypersurfaces, has some interesting and non-trivial features. For simplicity, in this section we consider a rational planar curve given by

$$\mathbf{p}(t) = \left(\frac{f(t)}{h(t)}, \frac{g(t)}{h(t)} \right), \tag{2.6}$$

for univariate polynomials f, g, h with $\text{GCD}(f, g, h) = 1$. However, the results are generalisable to higher dimensions. Using Lagrange interpolation as an implicitisation method can essentially proceed in one of two ways, which we describe below.

The first method, which we say uses *on-curve data*, looks for a non-trivial polynomial that vanishes at sufficiently many points along the parametric hypersurface.

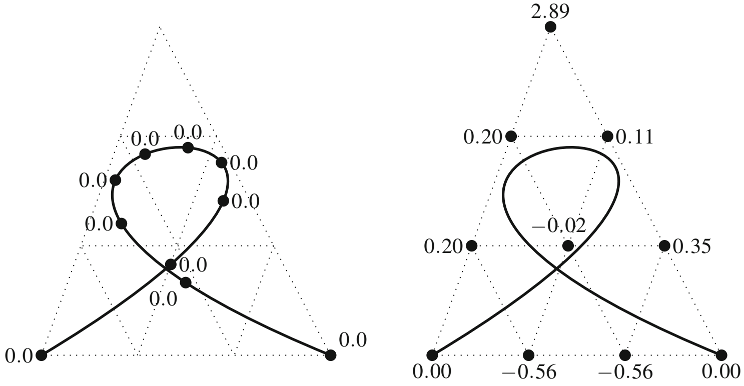


Fig. 2.5 Interpolation using parameter-uniform on-curve data (*left*) and uniform off-curve data (*right*). The basis for the implicit polynomial is a Bernstein basis over barycentric coordinates of a bounding triangle

This is described in [6] as implicitisation in the *Lagrange* basis, and is consistent with the general framework of Sect. 2.3.1. This is pictured, for the case of planar curves, on the left-hand side of Fig. 2.5. The number of nodes, N, required for exact implicitisation of a degree n rational curve is given in [6] by

$$N = n^2 + 1.$$

When the degree chosen for interpolation is smaller than required for exact implicitisation, this approach will produce approximations.

The second method, which we say uses *off-curve data*, is the method described by Marco and Martinez [27, 28]. This is pictured, for the case of planar curves, on the right-hand side of Fig. 2.5. This approach uses a resultant of the following two polynomials:

$$xh(t) - f(t), \; yh(t) - g(t).$$

The resultant is then evaluated at numerically at sufficiently many sample points $(x_i, y_j)_{i,j=0,0}^{n_1,n_2}$ in order to generate enough data to solve a bivariate Lagrange interpolation problem. Essentially, the data generated defines the implicit polynomial in a Lagrange basis and the method proceeds to generate the monomial coefficients by using a basis transformation matrix (i.e., via a Vandermonde matrix). Computing the determinant of a numerical matrix is much less computationally intensive than that of a symbolic matrix. One reason for this is that for numerical matrices, LU-decomposition is possible, where the matrix is decomposed into the product of a lower and an upper triangular matrix. The determinant is then given by the product of the entries on the diagonals of the matrices. Note that the approach of Marco and Martinez can be used with any type of resultant matrix.

For a given curve, an example of the two methods is shown in Fig. 2.5. Using the on-curve method, the interpolation data is all known to be zero (since it lies on the curve) but we evaluate the parametric curve at nodes in the parameter domain, in order to find the interpolation nodes in $\mathbb{R}^2$. The implicit coefficients are then given by the solution to a homogeneous linear system. In the off-curve method, the nodes are predetermined (e.g., on a uniform grid), but we have to evaluate the resultant in order to generate data for the interpolation. Both methods are also suitable for implicitising general hypersurfaces, although the off-curve method relies on suitable multivariate resultants.

For exact implicitisation using the on-curve method, care needs to be taken to ensure that the interpolation points lie in *general position*, so that the unique implicit polynomial can be found. It is well known that any five points in the plane, with no four collinear, determine a conic. This is because there are five degrees of freedom in the implicit representation of a conic (in fact, there are six coefficients, but since implicit representations are unchanged by non-zero scalar multiplication, one degree of freedom is removed). The equivalent statement for cubics is complicated by the famous Cayley-Bacharach theorem from algebraic geometry [16]:

Theorem 1 (Cayley-Bacharach theorem) *Consider the nine points in which two implicit planar cubic curves, I_1 and I_2, intersect. Then any cubic curve I_3 that passes through any eight of these points, also passes through the ninth.*

When sampling from a rational parametric curve, problems with finding points in general position can be avoided by a small oversampling [6]. This is the reason we use ten interpolation points, rather than nine, in the example of Fig. 2.5.

Both the on-curve and off-curve methods can also be used for approximate implicitisation. In general, the on-curve method provides better approximations than the off-curve method. In fact, the on-curve method, which approximates in the parameter domain, exhibits the same high convergence rates as the general approach of Sect. 2.3.1. For the off-curve method, the approximation must be made in the geometric domain, which generally results in greater errors. A discussion of how the Lebesgue constant from approximation theory is important in judging the quality of implicit approximations is presented in [6].

Hermite interpolation using implicit surfaces has also been investigated by several authors; see for example [1, 2]. These methods attempt to interpolate both position and derivative data, in order to obtain C^k-continuous curves or surfaces. Such requirements can also be met using approximate implicitisation, by adding constraints to the approximation [13]. The constraints can be added before or after approximation, the former via Lagrangian multipliers, and the latter by merging combinations of lower order approximations [15, 39].

2.3.4 Sparse Implicitisation

The main reason that multivariate polynomials of large total degree are computationally unattractive, is that any basis for such polynomials will have very large dimension. However, there will often exist a basis for the implicit description where several, or even most of the coefficients are zero. If it is known, a priori, which of these coefficients are zero, the complexity of the implicitisation problem can be reduced; in some cases substantially [17, 18].

In [18], techniques are introduced that exploit sparsity in the monomial basis representation of the implicit equations. For example, assume we want to implicitise a parabola defined by the control points $(-1,1)^T, (0,-1)^T$ and $(1,1)^T$, with all weights equal to one. The implicit equation of this curve is simply $y - x^2 = 0$, however, without this knowledge, we would generally proceed to use a basis of all six monomials of up to total degree 2 $\{1, x, y, x^2, xy, y^2\}$, forming a matrix of dimensions 5×6. If it is known that only the monomials y and x^2 are required to define the implicit equation, the size of the linear system can be reduced to 2×2.

In order to predict the support of the implicit function (i.e., the set of monomials that have non-zero coefficients in the implicit polynomial), the method of Emiris uses toric elimination theory. The method constructs the Newton polygon directly from the parametric representation and the support is then taken to include all monomials in the convex hull of the Newton polygon. Support prediction methods are also well suited for approximate implicitisation [6, 17]. Whereas in approximate implicitisation we choose to reduce the total degree, techniques using support prediction can remove any other monomials from the basis, even those that are not necessarily of high degree.

One of the main problems with using sparsity in practice is that freeform hypersurfaces do not, in general, exhibit a high degree of sparsity in the monomial basis. Also, since current algorithms only work in the monomial basis, basis transformations are required, which have potentially ill conditioned behaviour. In a sense, the basis functions described in Sects. 2.2.3 and 2.2.4, can be thought of as a sparse basis for performing implicitisation. However, since they depend on the control points, they are more complicated to construct than simple monomials. Further research is required to determine if suitable sparse bases can be defined for general freeform hypersurfaces.

In applications where exactness is required in order to create watertight models (such as in isogeometric analysis), sparse methods may prove a valuable tool in obtaining computationally tractable solutions.

2.4 Implicitisation of Envelope Curves, Surfaces and Hypersurfaces

In recent years, there has been an increase of interest in implicitisation of envelope curves and surfaces. This has particular applications in the field of robotics [32]. An envelope curve is defined as follows: consider a rational family of rational curves defined by

$$\mathbf{p}(s,t) = \left(\frac{p_1(s,t)}{g(s,t)}, \frac{p_2(s,t)}{g(s,t)} \right), \quad (s,t) \in [s_0, s_1] \times [t_0, t_1].$$

Either s or t can be thought of as a time-like parameter and the remaining parameter (either t or s) parametrises the curves. The envelope is a curve that touches all members of the family $\mathbf{p}$; two examples are pictured in Fig. 2.8 (p. 34). The envelope corresponds to the parameter values where the Jacobian becomes singular [32], and this definition can also be used to define envelopes in higher dimensions. A polynomial function $h(s,t)$, called the envelope function, can be defined in terms of the coefficients of $\mathbf{p}$ whose zero set corresponds exactly to such points. In general, the envelope does not have a simple parametrisation so the methods described earlier in this chapter are not directly relevant for envelope implicitisation. However, recently, Schulz and Jüttler showed that approximate implicitisation can be adapted to envelope curves in a similar way [32]. They showed that if q is the exact implicit representation of the envelope, then

$$q \circ \mathbf{p}(s,t) = \lambda(s,t) h(s,t)^2$$

for some unknown function λ. Thus, finding polynomials q and λ that approximate this equation gives methods for approximating the envelope.

A distinctive feature of envelope curves, like surfaces, is the high degrees that occur. Envelopes are also of interest in higher dimensions, where the problem of high degrees is exacerbated even further. Approximate implicitisation is therefore a potentially important tool in working with such manifolds.

In [7] a fast method for implicitising envelope curves, based on the method of [32] is defined. The method is also easily generalisable to higher dimensions, which is important for use in applications. For example, the possible positions of a robot with several connected arms is described by a parametric equation with one variable for each arm. In this way, high dimensions can easily occur, even with relatively simple geometries.

2.5 Applications of Methods for Implicitisation

The intention of this section is to highlight applications of (approximate) implicitisation algorithms. We consider three main applications: intersection algorithms, visualisation and robotics. We also discuss some open problems that have arisen by using the methods in practice.

2.5.1 Intersection Algorithms in Computer Aided Design

Intersection algorithms in CAD were the principal motivation behind the development of approximate implicitisation [13]. Whereas transversal intersections can be successfully approximated using subdivision schemes, tangential (and near tangential) intersections pose a more difficult problem [35]. For such cases, approximate implicitisation can be helpful [13, 39]. Algorithms for surface trimming are also related to intersection algorithms, and can benefit from the explicit implicitisation method outlined in [4].

2.5.1.1 Self-Intersections

There exist several methods for computing self-intersections of rational curves and surfaces, including [9, 29, 39]. In this section we mainly focus on the methods in [39], which are well suited to approximate implicitisation.

For an implicitly defined hypersurface q, we refer to points $\mathbf{p}$ where

$$q(\mathbf{p}) = 0 \quad \text{and} \quad \nabla q(\mathbf{p}) = \mathbf{0}$$

as singularities of the hypersurface. All self-intersections of algebraic hypersurfaces are singularities, but the converse is not necessarily true. For example, planar curves can also exhibit singularities as cusps or isolated points known as acnodes.

The simplest examples of self-intersections are for low degree curves. Clearly, in $\mathbb{R}^2$, a line will never intersect itself. Conic sections will also never exhibit singularities unless they can be decomposed into two lines (i.e., the implicit polynomial is reducible). For rational cubic curves with real coefficients, there always exists a real singularity, in the form of a self-intersection (crunode), cusp or acnode. For algebraic curves of degree 3 and higher there also exist non-singular curves, which do not have rational parametrisations [33]. Such curves have genus[4] greater than or equal to one, but these cases are not treated here; we concentrate on the case of rational curves, with genus zero. In [4] an explicit formula for the location of the singularity of a planar cubic curve in terms of a barycentric combination of its control points is presented. Simple conditions for when the singularity defines a self-intersection, and in particular, an *unwanted self-intersection* are also defined. In addition, the parameter values of the singularity can be computed explicitly as the roots of a special quadratic polynomial.

For most higher degree curves and surfaces, explicit formulas are not generally available and we often resort to numerical rootfinding. In [39], a method for computing self-intersections using approximate implicitisation is presented. Consider a given parametric hypersurface $\mathbf{p}(\mathbf{t})$, its normal $\mathbf{n}(\mathbf{t})$ and its (approximate) implicit representation q. The roots of the polynomial

$$\nabla q \circ \mathbf{p}(\mathbf{t}) \cdot \mathbf{n}(\mathbf{t}), \tag{2.7}$$

are then candidates for self-intersections [39].

[4] The genus of an algebraic curve is a concept from algebraic geometry that describes the topological properties of the curve.

It is normally possible to find exact implicit representations of rational parametric curves, since the degrees involved are not excessively high. For surfaces, exact methods are generally too complex, due to the high degrees that appear. Thomassen proposes using approximate implicitisation for computing self-intersections [39]. In the case of surfaces, Thomassen's approach to computing self-intersections gives excellent numerical results, although some post-processing is required to remove false positive intersections.

There exist alternative methods for finding self-intersections using resultants, which work for any degree [9, 11]. However, these methods are better suited to symbolic computation. Alternative methods for computing surface self-intersections numerically are presented in [29].

2.5.1.2 Curve and Surface Intersection via Algebraic Substitution

Computing intersections between two general freeform parametric curves or surfaces is generally much more complex than computing self-intersections. For example, whereas a planar cubic curve can only intersect itself once, two cubic curves can intersect in up to nine points in the plane. This can be generalised to higher degrees by the classical result known as Bézout's theorem:

Theorem 2 (Bézout's theorem) *Let $q_1(x,y) : \mathbb{R}^2 \to \mathbb{R}$ and $q_2(x,y) : \mathbb{R}^2 \to \mathbb{R}$ be irreducible polynomials of total degree m_1 and m_2 that define two algebraic curves. Then the total number of intersections between the curves (including complex intersections, intersections of higher multiplicity than one, and intersections at infinity) is given by $m_1 m_2$.*

Again, there are many methods for computing general intersections including both algebraic and subdivision approaches [35]. While subdivision methods perform well and give stable results for transversal intersections, tangential intersections pose more of a problem. In such cases, exact or approximate implicit representations can be useful.

For the sake of generality, in this section we outline the method for hypersurface/hypersurface intersections. This method utilises both implicit and parametric representations for computing intersections [13, 35]. Suppose we have two rational parametric hypersurfaces $\mathbf{p}_1(\mathbf{s}) : \boldsymbol{\Omega}_1 \to \mathbb{R}^d$ and $\mathbf{p}_2(\mathbf{t}) : \boldsymbol{\Omega}_2 \to \mathbb{R}^d$ for $\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2 \subset \mathbb{R}^{d-1}$. Suppose we also have corresponding implicit representations $q_1(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ and $q_2(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$. In computing the intersection, we are interested in finding the parameter values $\mathbf{s} \in \boldsymbol{\Omega}_1$ and $\mathbf{t} \in \boldsymbol{\Omega}_2$ such that $\mathbf{p}_1(\mathbf{s}) = \mathbf{p}_2(\mathbf{t})$. We are also interested in the set

$$P = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{x} = \mathbf{p}_1(\mathbf{s}) = \mathbf{p}_2(\mathbf{t}), \mathbf{s} \in \boldsymbol{\Omega}_1, \mathbf{t} \in \boldsymbol{\Omega}_2\}.$$

The $\mathbf{s}$-preimage of the intersection is defined as

$$S = \{\mathbf{s} \in \boldsymbol{\Omega}_1 : q_2(\mathbf{p}_1(\mathbf{s})) = 0\}.$$

A subset of the points $\mathbf{s} \in S$ is normally computed numerically. For each point $\mathbf{s}_i$, thus computed, it must be checked that the $\mathbf{t}$-preimage of the point $\mathbf{p}_1(\mathbf{s}_i)$ is in the domain $\boldsymbol{\Omega}_2$. Sederberg and Parry suggest using inversion formulas for this purpose [35]. It is also possible to use the reverse procedure of computing

$$T = \{\mathbf{t} \in \boldsymbol{\Omega}_2 : q_1(\mathbf{p}_2(\mathbf{t})) = 0\}.$$

However, this approach may be slower, since each pair of points in the two domains would need to be compared [39].

It is also possible to use approximate implicit representations in place of q_1 and q_2, in a direct analogue of the above procedure. This was proposed in Dokken's thesis [13].

One aspect of computing intersections, which is not taken into account when using approximate implicitisation, is that of topological consistency. This is concerned with the intersection results having compatible definitions in the different domains. According to [37], topological consistency requires that the representations of the intersection in the two parameter domains and the representation in $\mathbb{R}^d$, should all correspond to the same manifold. In general, the parameter domain preimage of hypersurface intersections will not be rational. For the case of surfaces, Song et al. [37] propose a linear perturbation method, whereby the parametric surfaces are altered slightly in order to force the intersection curve to be rational. In general, *approximate* implicit methods will not give topologically consistent results. However, this would be an interesting direction for future research.

2.5.1.3 Surface Trimming

Although piecewise rational surfaces are the predominant surface representation in CAD, often an additional operation is used to bound the region in which such a surface is defined. This process, known as surface trimming, defines regions of the parameter domain that correspond either to valid or invalid[5] surface points. The curves in the parameter domain are often expressed as densely sampled piecewise linear curves [19]. One reason for using piecewise linear curve data is that the problem of computing whether the point is considered valid or invalid is simplified, by counting ray intersections. However, such methods have limited accuracy.

The data for defining trimming curves may be generated, for example, from the intersection of two surfaces. This may be in the form of points in the parameter domain, which can be interpolated or approximated, and may also include derivative data. For example, a common approach is cubic Hermite interpolation, where we find a cubic curve $\mathbf{p}$, such that

[5] We refer to points on the original surface as valid if they also belong to the trimmed surface. Otherwise the points are invalid.

$$\mathbf{p}(0) = \mathbf{p}_0, \ \mathbf{p}(1) = \mathbf{p}_1, \ \mathbf{p}'(0) = \mathbf{m}_0, \text{ and } \mathbf{p}'(1) = \mathbf{m}_1,$$

for given point and derivative data $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{m}_0$ and $\mathbf{m}_1$. The rational Bézier form of cubic curves is particularly useful for this task, since interpolation and tangency constraints at the endpoints are given naturally by the control polygon [19]. In fact, the control points of such a polynomial cubic curve can be given explicitly as

$$\mathbf{c}_0 = \mathbf{p}_0, \ \mathbf{c}_1 = \mathbf{p}_0 + \frac{\mathbf{m}_0}{3}, \ \mathbf{c}_2 = \mathbf{p}_1 + \frac{\mathbf{m}_1}{3} \text{ and } \mathbf{c}_3 = \mathbf{p}_1.$$

Because the parametric form of cubic curves is not optimal for deciding which points lie inside or outside a curve, it is not a good choice for trimming curves. However, since we are given the control points and weights of the curve, the method of [4] can be used directly (i.e., no computationally expensive implicitisation methods are necessary). Hence, integration with existing cubic Hermite interpolation schemes is immediate, and will result in a piecewise implicit cubic representation. Moreover, it will provide a robust method for deciding whether or not the point lies within the trimming curve.

For implicitly representing piecewise curves in this way, some post-processing will be required in order to define the correct domains for each segment. Typically, restricting each segment to the convex hull of its control polygon will suffice. However, we must also define what happens outside the union of the convex hulls. One approach could be to use a Delaunay triangulation, similar to the method used in [25] for curve rendering.

2.5.2 Rendering Curves and Surfaces

Traditionally, the parametric form of B-spline and Bézier curves and surfaces has been used for rendering, since it is very easy to evaluate points lying on the curve or surface. Recently, real-time rendering in the implicit form has generated increased attention, particularly using ray casting and ray tracing [26, 31]. Such implementations rely heavily on the efficiency of modern GPU hardware, which can process per-pixel computations using highly parallel architectures. Since these methods compute on a per-pixel basis, a major advantage is that the resolution is independent of the zoom level or the viewpoint. Resolution independent methods have also been applied to curve rendering [25].

For applications that involve rendering regions bounded by curves, such as font shading, the implicit form of rational cubic and conic curves can be used. In [25], conic and cubic curves are evaluated implicitly by projecting one of a set of canonical curves onto screen space, resulting in highly efficient computations. The technique used in [30] is to employ the implicitisation method of [20], along with subdivisions to obtain a simple Bézier arch. In a similar way, the method of [4] could also be implemented for efficient rendering of cubic curves on the GPU.

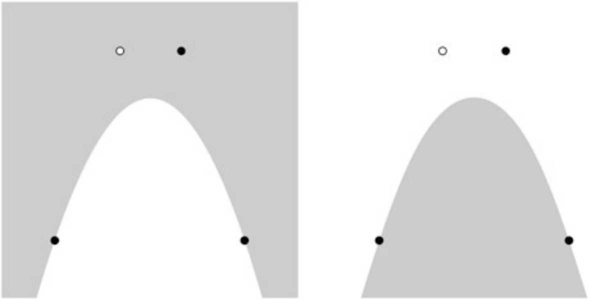


Fig. 2.6 An example of two cubic Bézier curves with control points $(0,0)$, (b_0,b_1), $(2/3,1)$ and $(1,0)$ (*left*: $(b_0,b_1) = (1/3+0.01,1)$, *right*: $(b_0,b_1) = (1/3-0.01,1)$, represented by *hollow circles*). The implicit representation flips orientation, despite only a small perturbation of the control points

A common technique for visualising implicit curves is to shade regions of the plane according to the sign of the implicit polynomial in those regions. For example, we can shade the region $W = \{(x,y) : q(x,y) \leq 0\}$ in white, and $G = \{(x,y) : q(x,y) > 0\} = W^c$ in grey (e.g., Fig. 2.6). However, such a method has some disadvantages in practice. Figure 2.6 shows two cubic Bézier curves rendered with such a method. The control points of these curves lie very close to each other, and yet the curve has flipped orientation. The reason for the change of orientation is that the curve has 'passed through' a conic section (both curves are *strictly cubic*, but they lie very close to a conic section).[6] This is not an artifact of the specific algorithm; it is in the nature of the implicit representation of cubic curves. This is necessarily true since it is also possible to transform between the two curves smoothly, without passing through the conic. This example highlights the unstable nature of cubic curves near conic sections. In [25], Loop and Blinn overcome such problems by adding a test to check that the 'inside' of the curve always lies to the right of the curve, in the direction of increasing parameter. We describe here an alternative technique that also solves other problems with unwanted self-intersections.

In addition to the implicit polynomial q, [4] gives explicit definitions of two lines, S_1 and S_2 that intersect each other at the double point, and intersect the curve at $\mathbf{c}_0$ and $\mathbf{c}_3$ respectively. As opposed to the 'inside/outside' approach, we propose defining the regions as follows:

$$W = \{(x,y) : S_1(x,y)q(x,y) \leq 0 \text{ OR } S_1(x,y)S_2(x,y) > 0\}, \quad G = W^c.$$

Using these definitions gives a smooth representation of the implicit curve, with no unwanted flips near conics. In addition, if an unwanted self-intersection occurs,

[6] The left figure has an acnode at $(1,100)$, and the right figure has a crunode at $(1,-100)$.

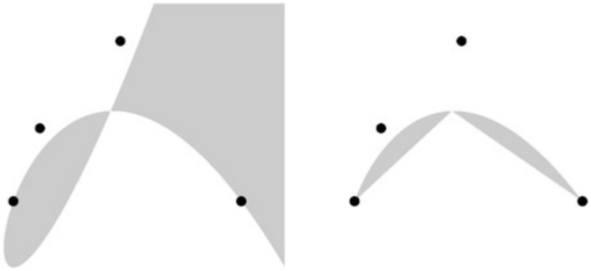


Fig. 2.7 An example of rendering a given cubic Bézier curve with the inside/outside method (*left*) and the proposed method (*right*). Note that no rootfinding is required. In the image to the *right*, it may be desirable to colour the triangle bounded by $\mathbf{c}_0$, $\mathbf{c}_3$ and the singularity in *grey*, but this is omitted for clarity

this method will automatically visually eliminate it (see Fig. 2.7). In previous algorithms, this has been achieved by subdividing the curves at the parameter values of the singularity [25, 30].

In a similar vein to the rendering of curves, the presence of the GPU has opened new doors to surface rendering in the implicit form [26, 31]. However, real-time ray tracing of implicit surfaces is quite severely restricted by the implicit degree and is currently only feasible for degrees approximately ≤ 12. Since a general bicubic parametric patch has implicit degree 18, the potential for exact methods appears to be limited.

The methods for approximate implicitisation presented earlier in this chapter have the potential to be coupled with implicit surface ray tracing methods. This would give the benefits of both the geometric control by manipulating the control polygon, and the high quality rendering that ray tracing produces. However, several challenges would need to be resolved.

An inherent problem with the implicit representation is the presence of extraneous branches. These branches define areas of the curve or surface that are not part of the region of interest. Although approximation has the potential to remove some extraneous branches, they are also a common occurrence in approximate implicitisation. One method for removing branches is to form linear combinations of several good approximations to the surface. When performing the algorithm for approximate implicitisation for a given degree m, we obtain a matrix (the right singular matrix of the SVD), that defines a basis for the space of polynomials of total degree m. The space is partitioned in such a way that the approximations (in the form of singular vectors), are ordered in quality by their corresponding singular values. If $\mathbf{v}_{\min}$ and $\mathbf{v}_{\min-1}$ refer to the singular vectors corresponding to the two smallest singular values, we can explore the linear space $\{\tau \mathbf{v}_{\min} + (1-\tau)\mathbf{v}_{\min-1}, \tau \in \mathbb{R}\}$ for an approximation in which extra branches are not present. If there are more good approximations, more dimensions can be added to the linear space. In particular, the orthogonal basis methods introduced in [6] generally exhibit a larger family of good approximations than the original method [13]. However, since criteria for when unwanted branches occur are hard to determine, it is difficult to automate this process. Moreover, though likely, it is not guaranteed that the linear space will contain a solution free from extra branches.

Defining suitable boundaries in which to render is another challenge that is more difficult to overcome for surfaces than curves. For surfaces, a 3D domain must be chosen for rendering. In certain cases, it may suffice to define the surface to lie within the 3D box that is limited by the upper and lower bounds of the control points, in a Cartesian system. However, this is not the case in general; the four boundary curves of a general bicubic patch can define any cubic space curve. Moreover, these boundary curves do not always define the silhouette of the surface. The method in [7] for computing envelopes, is related to the problem of finding silhouettes. Utilising this for the case of rendering may be a direction for future research, although performance would be a major obstacle with current hardware and implementations.

2.5.3 Robotics

In [7] an extension to the method for approximate implicitisation of envelope curves first published in [32] is proposed. Envelope curves have a variety of applications in robotics, including defining boundaries, collision detection and gearing. Envelope implicitisation is also interesting from a theoretical point of view, in giving an explicit definition to the curve.

In [32] a method for piecewise implicit approximation of envelope curves is presented along with several examples. One reason for choosing piecewise approximation is that the method becomes very computationally expensive for all but the lowest degrees. The new implementation in [7] allows faster approximations, thus in this section we present examples of higher degree implicitisations, as opposed to piecewise approximations.

In Fig. 2.8 we show the method applied to two different families of curves, both of which are implicitised at degree 6. The first, which is a quadratic family of circles of variable radius, has the homogeneous definition

$$\mathbf{p}(s,t) = \sum_{i=0}^{2}\sum_{j=0}^{4}(x_{ij}, y_{ij}, w_{ij})B_i^2(s)B_j^4(t),$$

where

$$(x_{ij})_{i,j=(0,0)}^{(2,4)} = \begin{pmatrix} 1/3 & -1/25 & 1/9 & 1/25 & 1/3 \\ 1/3 & -9/100 & 1/9 & 9/100 & 1/3 \\ 2/3 & -7/50 & 2/9 & 7/50 & 2/3 \end{pmatrix},$$

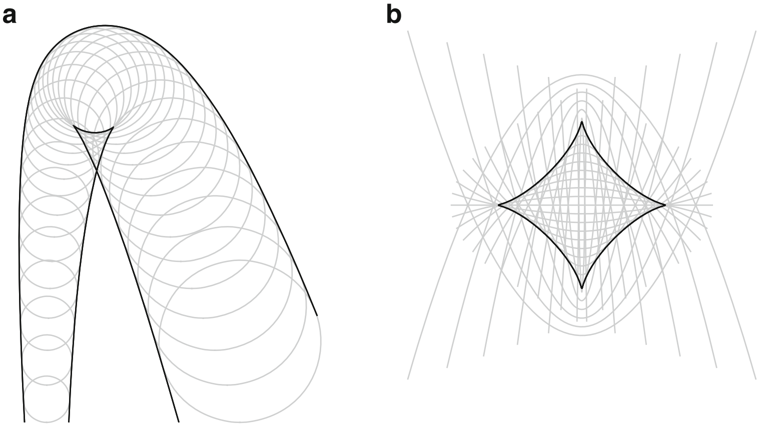


Fig. 2.8 The envelope curves defined by the families of curves given in Sect. 2.5.3 and computed using the techniques of [7]. The implicit representation has degree 6 in both cases

$$(y_{ij})_{i,j=(0,0)}^{(2,4)} = \begin{pmatrix} 0 & 0 & 4/75 & 0 & 0 \\ 1 & 0 & 34/75 & 0 & 1 \\ 0 & 0 & 14/75 & 0 & 0 \end{pmatrix},$$

$$(w_{ij})_{i,j=(0,0)}^{(2,4)} = \begin{pmatrix} 1 & 0 & 1/3 & 0 & 1 \\ 1 & 0 & 1/3 & 0 & 1 \\ 1 & 0 & 1/3 & 0 & 1 \end{pmatrix}.$$

The seemingly higher degree in the t-parameter direction is because the degree has been raised in order to obtain a better parametrisation of the circles. The family is really biquadratic in nature. For the implicit degree of the envelope curve we take $m = 6$. However, envelope implicitisation using this method also requires a choice of bidegree for the coupling function λ, which we denote by (k_1, k_2). In order to obtain good results for this example, we require $k_1 = 10$ and $k_2 = 16$. This particular choice of bidegree (k_1, k_2), is discussed in more detail in [7]. The necessity of using such high degrees makes the algorithm rather slow, but the example shows that accurate implicitisations of moderately high degree are possible. Also, notice that the envelope function, $h(s,t)$, has three branches within the region of interest. Thus, attempting to implicitise the entire envelope with a single polynomial is a relatively complicated problem. If a piecewise implementation is used, these branches would normally be considered separately. This example is generated by a medial axis transform of the type which appear commonly in robotics. Of course, in this example, a square root parametrisation of the envelope curve is possible, thus less computationally intensive methods could be used by generating point data on the curve.

The second example is a quadratic family of parabolas, defined by

$$(x_{ij})_{i,j=(0,0)}^{(2,4)} = \begin{pmatrix} 1/3 & -1/25 & 1/9 & 1/25 & 1/3 \\ 1/3 & -9/100 & 1/9 & 9/100 & 1/3 \\ 2/3 & -7/50 & 2/9 & 7/50 & 2/3 \end{pmatrix},$$

$$(y_{ij})_{i,j=(0,0)}^{(2,4)} = \begin{pmatrix} 0 & 0 & 4/75 & 0 & 0 \\ 1 & 0 & 34/75 & 0 & 1 \\ 0 & 0 & 14/75 & 0 & 0 \end{pmatrix},$$

with the weight function $w(s,t) \equiv 1$.

Using a single polynomial implicit representation of the envelope rather than a piecewise approximation can be somewhat advantageous. Since we only need a single polynomial for the entire region of interest, there are none of the complications of defining 2D regions for the pieces. In addition, for higher degrees, the approximation can be expected to be somewhat better, due to the high convergence rates computed in [7].

Acknowledgements The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement no. PITN-GA-2008-214584 (SAGA), and from the Research Council of Norway (IS-TOPP).

References

1. C.L. Bajaj, I. Ihm, Algebraic surface design with Hermite interpolation. ACM Trans. Graph. **11**(1), 61–91 (1992)
2. C.L. Bajaj, I. Ihm, J. Warren, Higher-order interpolation and least-squares approximation using implicit algebraic surfaces. ACM Trans. Graph. **12**(4), 327–347 (1993). doi:10.1145/159730.159734
3. O.J.D. Barrowclough, Approximate methods for change of representation and their applications in CAGD, Ph.D. thesis, University of Oslo, 2013
4. O.J.D. Barrowclough, A basis for the implicit representation of planar rational cubic Bézier curves. Comput. Aided Geom. Des. **31**, 48–167 (2014). doi:10.1016/j.cagd.2014.02.007
5. O.J.D. Barrowclough, T. Dokken, Approximate implicitization of triangular Bézier surfaces, in *Proceedings of the 26th Spring Conference on Computer Graphics, SCCG'10*, Budmerice (ACM, New York, 2010), pp. 133–140. doi:10.1145/1925059.1925084
6. O.J.D. Barrowclough, T. Dokken, Approximate implicitization using linear algebra. J. Appl. Math. (2012). doi:10.1155/2012/293746
7. O.J.D. Barrowclough, B. Jüttler, T. Schulz, Fast approximate implicitization of envelope curves using Chebyshev polynomials, in *Latest Advances in Robot Kinematics*, ed. by J. Lenarčič, M. Husty (Springer, Dordrecht, 2012), pp. 205–212. doi:10.1007/978-94-007-4620-6_26
8. Busé, L.: Implicit matrix representations of rational Bézier curves and surfaces. Compu. Aided Des. **46**(0), 14–24 (2014). doi:10.1016/j.cad.2013.08.014. 2013 SIAM Conference on Geometric and Physical Modeling
9. L. Busé, C. D'Andrea, Singular factors of rational plane curves. J. Algebra **357**, 322–346 (2012). doi:10.1016/j.jalgebra.2012.01.030
10. F. Chen, The μ-basis of a planar rational curve – properties and computation. Graph. Models **64**, 368–381 (2003). doi:10.1016/S1077-3169(02)00017-5

11. E.W. Chionh, T.W. Sederberg, On the minors of the implicitization Bézout matrix for a rational plane curve. Comput. Aided Geom. Des. **18**, 21–36 (2001)
12. R.M. Corless, M. Giesbrecht, I. Kotsireas, S. Watt, Numerical implicitization of parametric hypersurfaces with linear algebra, in *Artificial Intelligence and Symbolic Computation* (Springer, Berlin/New York, 2001), pp. 174–183
13. T. Dokken, Aspects of intersection algorithms and approximation, Ph.D. thesis, University of Oslo, 1997
14. T. Dokken, Approximate implicitization, in *Mathematical Methods for Curves and Surfaces* (Vanderbilt University Press, Nashville, 2001), pp. 81–102
15. T. Dokken, J.B. Thomassen, Weak approximate implicitization, in *IEEE International Conference on Shape Modeling and Applications, 2006 (SMI 2006)*, Matsushima (IEEE, 2006)
16. D. Eisenbud, M. Green, J. Harris, Cayley-Bacharach theorems and conjectures. Bull. Am. Math. Soc. **33**(03), 295–325 (1996). doi:10.1090/S0273-0979-96-00666-0
17. I.Z. Emiris, T. Kalinka, C. Konaxis, Implicitization of curves and surfaces using predicted support, in *Proceedings of the 2011 International Workshop on Symbolic-Numeric Computation (SNC'11)*, San Jose, 2011, pp. 137–146. doi:10.1145/2331684.2331705
18. I.Z. Emiris, I.S. Kotsireas, Implicitization exploiting sparseness, in *Geometric and Algorithmic Aspects of Computer-Aided Design and Manufacturing*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 67 (American Mathematical Society, Providence, 2005), pp. 281–298
19. G. Farin, *Curves and Surfaces for CAGD: A Practical Guide* (Morgan Kaufmann, San Francisco, 2002)
20. M.S. Floater, Rational cubic implicitization, in *Mathematical Methods for Curves and Surfaces* (Vanderbilt University Press, Nashville, 1995), pp. 151–159
21. X.S. Gao, S.C. Chou, Implicitization of rational parametric equations. J. Symb. Comput. **14**, 1–15 (1992)
22. C.M. Hoffmann, Implicit curves and surfaces in CAGD. IEEE Comput. Graph. Appl. **13**(1), 79–88 (1993). doi:10.1109/38.180121
23. B. Jüttler, A. Felis, Least-squares fitting of algebraic spline surfaces. Adv. Comput. Math. **17**(1), 135–152 (2002)
24. I.S. Kotsireas, Panorama of methods for exact implicitization of algebraic curves and surfaces, in *Geometric Computation*, ed. by F. Chen, D. Wang. Lecture Notes Series on Computing, vol. 11, chap. 4 (World Scientific, Singapore, 2004), pp. 126–155
25. C.T. Loop, J. Blinn, Resolution independent curve rendering using programmable graphics hardware. ACM Trans. Graph. (TOG) **24**, 1000–1009. (ACM, 2005)
26. C.T. Loop, J. Blinn, Real-time GPU rendering of piecewise algebraic surfaces. ACM Trans. Graph. **25**(3), 664–670 (2006). doi:10.1145/1141911.1141939
27. A. Marco, J.J. Martinez, Using polynomial interpolation for implicitizing algebraic curves. Comput. Aided Geom. Des. **18**(4), 309–319 (2001). doi:10.1016/S0167-8396(01)00033-4
28. A. Marco, J.J. Martinez, Implicitization of rational surfaces by means of polynomial interpolation. Comput. Aided Geom. Des. **19**(5), 327–344 (2002). doi:10.1016/S0167-8396(02)00094-8
29. D. Pekerman, G. Elber, M.S. Kim, Self-intersection detection and elimination in freeform curves and surfaces. Comput. Aided Des. **40**(2), 150–159 (2008). doi:10.1016/j.cad.2007.10.004
30. R. Pfeifle, Rendering cubic curves on a GPU with Floater's implicitization. J. Graph. Tools **16**(2), 105–122 (2012). doi:10.1080/2151237X.2012.653284
31. M. Reimers, J.S. Seland, Ray casting algebraic surfaces using the frustum form. Comput. Graph. Forum **27**(2), 361–370 (2008)
32. T. Schulz, B. Jüttler, Envelope computation in the plane by approximate implicitization. Appl. Algebra Eng. Commun. Comput. **22**(4), 265–288 (2011). doi:10.1007/s00200-011-0149-1
33. T.W. Sederberg, D.C. Anderson, R.N. Goldman, Implicit representation of parametric curves and surfaces. Comput. Vis. Graph. Image Process. **28**(1), 72–84 (1984)

34. T.W. Sederberg, F. Chen, Implicitization using moving curves and surfaces. in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles (ACM, New York, 1995), pp. 301–308. doi:10.1145/218380.218460
35. T.W. Sederberg, S.R. Parry, Comparison of three curve intersection algorithms. Comput. Aided Des. **18**(1), 58–63 (1986)
36. L. Shen, F. Chen, B. Jüttler, J. Deng, Approximate μ-bases of rational curves and surfaces, in *Geometric Modeling and Processing-GMP 2006*, Pittsburgh, 2006, pp. 175–188
37. X. Song, T.W. Sederberg, J. Zheng, R.T. Farouki, J. Hass, Linear perturbation methods for topologically consistent representations of free-form surface intersections. Comput. Aided Geom. Des. **21**(3), 303–319 (2004)
38. M.C. Stone, T.D. DeRose, A geometric characterization of parametric cubic curves. ACM Trans. Graph. **8**(3), 147–163 (1989). doi:10.1145/77055.77056
39. J.B. Thomassen, Self-intersection problems and approximate implicitization, in *Computational Methods for Algebraic Spline Surfaces* (Springer, Berlin/New York, 2005), pp. 155–170
40. E. Wurm, B. Jüttler, Approximate implicitization via curve fitting, in *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, Aachen (Eurographics Association, 2003), pp. 240–247

Chapter 3
Sparse Implicitization via Interpolation

Ioannis Z. Emiris, Tatjana Kalinka, and Christos Konaxis

3.1 Introduction

Implicitization is the process of changing the representation of a geometric object from parametric to algebraic, or implicit. It is an important operation in Algebraic Geometry with applications in Computer Aided Design (CAD) and Geometric Modeling. There have been numerous approaches, including those based on Gröbner bases [2, 6, 19], resultants [4, 22], residues [3], moving lines and surfaces [23], and μ-bases [8].

Our approach follows the standard method of interpolating the unknown coefficients of the implicit polynomial given a superset of its monomials. It can be applied to planar curves, surfaces, or hypersurfaces of any dimension, given by a polynomial, rational or trigonometric parameterization, including those with base points. It is well known that base points raise important issues for certain methods. Our method has its limits: in the case of trigonometric parameterizations they have to be convertible to rational functions and the support prediction step requires that geometric objects have to be presented in the monomial basis. On the upside, we have employed our approach to parameterizations in the Bernstein basis by converting them to the monomial basis, see Sect. 3.3.2.

The main ingredient of our method is the Newton polytope of the implicit equation, or *implicit polytope*.

Definition 3.1 Given a polynomial $f = \sum_a c_a t^a \in \mathbb{R}[t_1, \ldots, t_n]$, $t^a = t_1^{a_1} \cdots t_n^{a_n}, a \in \mathbb{N}^n, c_a \in \mathbb{R}$, its *support* is the set $\{a \in \mathbb{N}^n : c_a \neq 0\}$, where $\mathbb{N} = \{0, 1, \ldots\}$; its *Newton polytope* $N(f)$ is the convex hull of its support.

I.Z. Emiris • T. Kalinka • C. Konaxis (✉)
University of Athens, Athens, Greece
e-mail: emiris@di.uoa.gr; kalinkat@di.uoa.gr; ckonaxis@di.uoa.gr

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_3

There are several methods for computing the implicit polytope, such as those based on tropical geometry, or mixed fiber polytopes, see e.g. [9,16,28]. In this work the implicit polytope is computed from the Newton polytope of the sparse (or toric) resultant, or *resultant polytope*, of polynomials defined by the parametric equations [11–13]. Under certain genericity assumptions, the implicit polytope coincides with a projection of the resultant polytope, see [14, 15]. The rest of our approach does not depend on the method used to compute the implicit polytope. In fact, [28, sec.4] states that

> Knowing the Newton polytopes reduces computing the [implicit] equation to numerical linear algebra. The numerical mathematics of this problem is interesting and challenging [...]

The set of lattice points contained in the implicit polytope form a superset of the support of the implicit equation, or *implicit support*. This predicted support is used to build a numeric matrix, whose kernel is, ideally, 1-dimensional, thus yielding (up to a nonzero scalar multiple) the coefficients corresponding to the predicted implicit support. This is a standard case of *sparse interpolation* of the polynomial from its values. When dealing with hypersurfaces of high dimension, or when the support contains a large number of lattice points, then exact solving is expensive. Since the kernel can be computed numerically, our approach also yields an approximate sparse implicitization method.

The kernel of the numerical matrix may be of high dimension. We address this situation by presenting techniques that alleviate this phenomenon. More formally, we relate it to the geometry of the predicted support, which is a superset of the true implicit support. Another reason for obtaining a high-dimensional kernel is that the numeric evaluation of the support monomials may not be sufficiently generic.

It is possible to apply our method to a more general problem, namely to computing the discriminant of a multivariate polynomial, which is an important question with several geometric applications. The vanishing of the discriminant characterizes the existence of multiple roots of the given polynomial. This is a hard computation, since explicit formulas only exist for low-degree univariate polynomials. In general, one can reduce discriminant computation to computing the resultant of a rather large system, comprised of the polynomial and its partial derivatives, but this is inefficient. Instead, we reduce discriminant computation to sparse implicitization, thus obtaining an output-sensitive algorithm, whose complexity depends on the size of the discriminant's Newton polytope, see [15] for details. Moreover, this technique can be used to compute discriminants of well-constrained systems as well as resultants because the latter can be viewed as a special case of discriminants.

This chapter is organized as follows: Sect. 3.2 defines basic concepts, overviews existing work and sketches the framework of sparse implicitization. Section 3.3 presents our implicitization algorithm and its implementation, discusses the case of higher dimensional kernels and applications to Bézier and Non-uniform rational B-spline (NURBS) curves. We conclude in section "Conclusion and Future Work" and provide some directions for future work.

3.2 Implicitization Reduced to Elimination

In this section we describe how elimination theory, and in particular sparse elimination theory, can be used in implicitization by providing the implicit polytope, or a polytope that contains it.

Let us describe the problem of implicitization formally. A *parameterization* of a geometric object can be described by a set of parametric functions:

$$x_0 = f_0(t), \dots,\ x_n = f_n(t), \quad t := (t_1, t_2, \dots, t_n), \tag{3.1}$$

where t is the vector of parameters and $f := (f_0, \dots, f_n)$ is a vector of coordinate functions (continuous functions, polynomial, rational, or trigonometric, defined on some product $\Omega := \Omega_1 \times \cdots \times \Omega_n$, of one-dimensional intervals Ω_i, of values of $t_1, \dots, t_n$. We assume that trigonometric parameterizations may be converted to rational functions using the standard half-angle transformation

$$\sin\theta = \frac{2\tan\theta/2}{1+\tan^2\theta/2}; \quad \cos\theta = \frac{1-\tan^2\theta/2}{1+\tan^2\theta/2}.$$

The *implicitization problem* asks for the smallest algebraic variety containing the closure of the image of the parametric map $f : \mathbb{R}^n \to \mathbb{R}^{n+1} : t \mapsto f(t)$. This image is contained in the variety defined by the ideal of all polynomials p s.t. $p(f_0(t), \dots, f_n(t)) = 0, \ \text{for all } t \text{ in } \Omega$. We restrict ourselves to the case when this is a principal ideal, and we wish to compute its unique (up to scalar multiple) defining polynomial

$$p(x_0, \dots, x_n) = 0, \tag{3.2}$$

given its Newton polytope, or a polytope that contains it. We can regard the variety in question as the projection of the graph of the map f to the last $n+1$ coordinates. If f is polynomial, implicitization is reduced to eliminating t from the polynomial system

$$F_i := x_i - f_i(t) \in (\mathbb{R}[x_i])[t],\ i = 0, \dots, n,$$

seen as polynomials in t with coefficients which are functions of the x_i. This is also the case for rational parameterizations

$$x_i = \frac{f_i(t)}{g_i(t)},\ i = 0, \dots, n, \tag{3.3}$$

which can be represented as polynomials

$$F_i := x_i g_i(t) - f_i(t) \in (\mathbb{R}[x_i])[t],\ i = 0, \dots, n, \tag{3.4}$$

where we have to take into account that the $g_i(t)$ cannot vanish by adding the following polynomial to the system $F_{n+1} = 1 - yg_0(t) \cdots g_n(t)$, where y is a new variable. If one omits F_{n+1}, the generator of the corresponding (principal) ideal may be a multiple of the implicit equation. Then the extraneous factor corresponds to the g_i. Eliminating t, y may be done by taking the *resultant* of the polynomials in (3.4).

Let $A_i \subset \mathbb{Z}^n$, $i = 0, \dots, n+1$, be the supports of the polynomials F_i and consider the generic polynomials

$$F'_0, \dots, F'_n,\ F'_{n+1} \tag{3.5}$$

with the same supports A_i and symbolic coefficients c_{ij}.

Definition 3.2 The *sparse resultant* $\mathrm{Res}(F'_0, \dots, F'_{n+1})$ is a polynomial in the c_{ij} with integer coefficients, namely

$$R \in \mathbb{Z}[c_{ij} : i = 0, \dots, n+1,\ j = 1, \dots, |A_i|],$$

which vanishes if and only if the system $F'_0 = F'_1 = \cdots = F'_{n+1} = 0$ has a common root in a specific variety. This variety is the projective variety $\mathbb{P}^{n+1}$ over the algebraic closure of the coefficient field in the case of projective (or classical) resultants, or the toric variety defined by the A_i's.

The sparse resultant is unique up to sign.

The theory of the sparse resultant can be used to derive degree bounds for the implicit equation. For the case where the denominators of the parameterization are the same we have the following.

Proposition 3.1 *Consider the rational parameterization*

$$x_i = \frac{f_i(t)}{g(t)},\ i = 0, \dots, n, \quad t = (t_1, t_2, \dots, t_n). \tag{3.6}$$

Let $V \subset \mathbb{C}^{n+1}$ be the Zariski closure of the image of the parameterization and define polynomials $F_i := x_i g(t) - f_i(t) \in (\mathbb{C}[x_i])[t]$, $i = 0, \dots, n$, with supports A_i. Then, the total degree of the implicit polynomial is bounded by $n!$ times the volume of the convex hull of $A_0 \cup \cdots \cup A_n$. The degree of the implicit polynomial in each x_j, $j \in \{0, \dots, n\}$, is bounded by the mixed volume of the F_i, $i \neq j$, seen as polynomials in t.

Proof For simplicity we shall describe our arguments in the affine space instead of the projective space.

The degree of V is obtained as the number of points in the intersection of V with a generic linear subspace of dimension equal to codimension$(V) = 1$. Let $H_i(x_0, \dots, x_n) = a_{i,0}x_0 + \cdots + a_{i,n}x_n + a_{i,n+1} = 0$, $i = 1, \dots, n$, be n generic affine hyperplanes. Substituting the parametric expressions in (3.6) to the H_i and clearing the denominators, we obtain a system of n non-linear equations in the parameters t:

$$a_{i,0} f_0(t) + \cdots + a_{i,n} f_n(t) + a_{i,n+1} g(t) = 0, \; i = 1, \ldots, n. \tag{3.7}$$

The number of common roots of the system (3.7), hence, the number of intersection points of V with a generic linear subspace of dimension 1, is bounded by the mixed volume [7], of the Newton polytopes of the polynomials in (3.7). The support of each polynomial in (3.7) is the union of the supports A_i of the polynomials F_i, $i = 0, \ldots, n$. Hence, every polynomial has the same Newton polytope and the mixed volume of the system equals $n!$ times the volume of this Newton polytope.

Similarly, to bound the degree of the implicit equation in each x_j, $j = 0, \ldots, n$, intersect V with n lines $x_i = k_i$, $i = 0, \ldots, j-1, j+1, \ldots, n$, where k_i is a generic constant. Substituting the parametric expressions in (3.6) to each $x_i, i \neq j$, we obtain a system of n equations in the n variables t:

$$F_i = k_i g(t) - f_i(t) = 0, \; i = 0, \ldots, j-1, j+1, \ldots, n,$$

each having support A_i. Then, the mixed volume $\mathrm{MV}(A_0, \ldots, A_{j-1}, A_{j+1}, \ldots, A_n)$ of the system bounds the degree of the implicit equation in the variable x_j.

Our approach refines the above information since it is based on the Newton polytope of the implicit equation rather than degree bounds. In order to exploit sparseness in the implicit polynomial, the problem of computing the Newton polytope of a rational hypersurface in the framework of sparse elimination theory was first posed in [27] for generic Laurent polynomial parameterizations.

The implicit equation of the parametric hypersurface defined in (3.4) equals the resultant $\mathrm{Res}(F_0, \ldots, F_{n+1})$, provided that the latter does not vanish identically. Thus, the implicit equation can be obtained from $\mathrm{Res}(F_0', \ldots, F_{n+1}')$ by specializing the symbolic coefficients of the F_i''s to the actual coefficients of the F_i's, provided that this specialization is generic enough. In this case, the implicit polytope equals the resultant polytope projected to the space of the implicit variables, i.e. the Newton polytope of the specialized resultant, up to some translation. When this condition fails for the given specialization of the c_{ij}'s, the support of the specialized resultant is a superset of the support of the actual implicit polynomial modulo a translation. This follows from the fact that the method computes the same resultant polytope as the tropical approach, where the latter is specified in [26], see Proposition 3.2. Note that there is no exception even in the presence of base points.

Algorithms based on tropical geometry have been offered in [10, 26, 28]. This method computes the abstract tropical variety of a hypersurface parametrized by generic Laurent polynomials in any number of variables, thus yielding its implicit support. For non-generic parameterizations of rational curves, the implicit polygon is predicted. In [21], the authors describe efficient algorithms implemented in the GFan library for the computation of Newton polytopes of specialized resultants, which may then be applied to predict the implicit polytope.

Proposition 3.2 ([26, Prop.5.3]) *Let $f_0, \ldots, f_n \in \mathbb{C}[t_1^{\pm 1}, \ldots, t_n^{\pm 1}]$ be any Laurent polynomials whose ideal I of algebraic relations is principal, say $I = \langle p \rangle$, and let $P_i \subset \mathbb{R}^n$ be the Newton polytope of f_i. Then the resultant polytope which is* *constructed combinatorially from $P_0, \dots, P_n$ contains a translate of the Newton polytope of p.*

A direct consequence is that our method produces the precise implicit polytope when the input is sufficiently generic, otherwise it gives a polytope that contains the implicit polytope. Our method relies on sparse elimination theory: the implicit polytope is obtained from the projection of the resultant polytope of the polynomials in (3.5) defined by the specialization of their symbolic coefficients to those of the polynomials in (3.4).

In the case of curves, the implicit polytope is directly determined in [11]. In higher dimensions we use the algorithm developed in [13]. This is an incremental algorithm to compute the resultant polytope, or its orthogonal projection along a given direction. By the Cayley Trick [20], there exists a bijection from the set of regular triangulations of the Cayley pointset $\mathcal{C}$ of the Newton polytopes of the polynomials F_i', to the set $\mathcal{M}$ of regular tight mixed subdivisions of the Minkowski sum of these Newton polytopes. In [25] a surjection is established from the set $\mathcal{M}$ to the the vertices of the resultant polytope. Combining these results, one obtains a surjection from $\mathcal{C}$ to the vertices of the resultant polytope. The algorithm exploits this surjection and computes one triangulation of $\mathcal{C}$ per vertex and one per facet of projection of the resultant polytope. It is implemented in the package `ResPol` [17]. The algorithm exactly computes vertex- and halfspace-representations of the target polytope and it is output-sensitive. It is efficient for inputs relevant to implicitization: it computes the polytope of surface equations within 1 s, assuming there are less than 100 terms in the parametric polynomials, which includes all common instances in geometric modeling.

3.3 Algorithm and Implementation

In this section we present our implicitization algorithm and its implementation. We discuss the cases of higher dimensional kernels and the applicability of our method to Bézier and NURBS curves. We also compare our method against other methods, such as the Gröbner bases method and `Maple`'s native routines.

The main steps of our algorithm are the following:

Algorithm 1

```
Input: Polynomial or rational parameterization $x_i = f_i(t),\ i = 0, \dots, n$.
Output: Implicit polynomial $p(x_0, \dots, x_n)$ in the monomial basis in $\mathbb{N}^{n+1}$.

Step 1: Determine (a polytope containing) the implicit polytope.
Step 2: Compute all lattice points $S \subseteq \mathbb{N}^{n+1}$ in the polytope.
Step 3: Repeat $\geq |S|$ times: Select value $\tau$ for $t$, evaluate $x_i(\tau),\ i = 0, \dots, n$,
    thus evaluating each monomial with exponent in $S$. This yields a matrix $M$.
Step 4: Given the matrix $M$, solve $M\,\vec{p} = 0$ for kernel $\vec{p}$.
```

Step 5: Compute the greatest common divisor (GCD) of $\vec{p}_i^\top \cdot \vec{m},\ i = 1,\dots,\text{corank}(M)$, where $\vec{m}$ is the vector of monomials with exponent in S and $\vec{p}_i$ are the kernel vectors of M.

Step 6: Return the primitive part of the computed GCD.

Let us describe in more detail the construction of matrix M in Step 3 assuming that the parameterization is rational. Let $S := \{s_1,\dots,s_{|S|}\}$; each $s_j = (s_{j0},\dots,s_{jn})$ is an exponent of a (potential) monomial $m_j := x^{s_j} = x_0^{s_{j0}} \cdots x_n^{s_{jn}}$ of the implicit polynomial, where $x_i = f_i(t)/g_i(t)$. We evaluate m_j at some $\tau_k,\ k = 1,\dots,\mu$, avoiding values that make the denominators of the parametric expressions close to 0, and obtain $m_j|_{t=\tau_k} := \prod_i \left(\frac{f_i(\tau_k)}{g_i(\tau_k)}\right)^{s_{ji}}$. Thus, we build an $\mu \times |S|,\ \mu \geq |S|$, matrix M with rows indexed by the τ_k and columns indexed by the m_j:

$$M = \begin{bmatrix} m_1|_{t=\tau_1} & \cdots & m_{|S|}|_{t=\tau_1} \\ \vdots & \cdots & \vdots \\ m_1|_{t=\tau_\mu} & \cdots & m_{|S|}|_{t=\tau_\mu} \end{bmatrix}.$$

We compute the kernel of the matrix M either symbolically or numerically. For exact computations we prefer `Maple`, while for the numerical ones `SAGE` using Singular Value Decomposition (SVD). In our `Maple` implementation the computation of the lattice points in Step 2 is done, for up to 4 dimensions, by routines that utilize the `Maple` package `convex` [18], whereas our `SAGE` implementation uses its built-in functions for the same task. For higher dimensions we have employed the software package `Normaliz`.

When the kernel computation in Step 4 is done numerically, we build a rectangular overconstrained matrix M, by evaluating the monomials in S at more than $|S|$ values, in order to increase the numerical stability.

Experiments with curves and surfaces in the monomial basis, as well as in the Bernstein basis, show that when building the matrix M, it is important to choose τ values that are suitable for the specific instance.

Choosing τ for implicitization of classical algebraic curves and surfaces, we have experimented with random integers in the range $-\mu^2 \dots \mu^2$, random rational numbers, complex μ-th roots of unity and random complex numbers modulo 1. Random integers offer the most numerically stable results, however with large matrices they result in fast growth of matrix entries. Random rational values have proved to be unreliable when implicitizing classical algebraic curves and surfaces, although complex values are numerically stable. The case of curves and surfaces in the Bernstein basis is treated in Sect. 3.3.2.

By the construction of the matrix M using values τ that correspond to points on the parametric surface, we have the following:

Lemma 3.1 *Any polynomial in the basis of monomials indexing M, with coefficient vector in the kernel of M, is a multiple of the implicit polynomial p.*

3.3.1 Multidimensional Kernel

In this subsection we address the case of high dimensional kernels, we relate this situation to the geometry of the predicted support and propose techniques that alleviate this phenomenon.

Let Q, P be the predicted and actual implicit polytopes respectively. When Q is significantly larger than P, we obtain a matrix M of corank > 1, see Theorem 3.1 below.

Example 3.1 Consider the paraboloid $x_0^2 + x_1^2 - x_2 = 0$. If its parameterization is

$$x_0 = u,\ x_1 = v,\ x_2 = u^2 + v^2,$$

then we define the polynomials $F_0 = x_0 - u,\ F_1 = x_1 - v,\ F_2 = x_2 - u^2 - v^2$, with supports $A_0 = \{(1,0),\ (0,0)\},\ A_1 = \{(0,1),\ (0,0)\},\ A_2 = \{(2,0),\ (0,2),\ (0,0)\}$. The Cayley pointset of these supports is the set

$$\begin{aligned}\mathcal{C} =&\{(1,0,0,0),\ (0,0,0,0), (0,1,1,0), (0,0,1,0), (2,0,0,1),\\ &(0,2,0,1), (0,0,0,1)\}.\end{aligned}$$

The predicted implicit polytope Q has vertices $(2,0,0), (0,2,0), (0,0,1)$ which are obtained from some triangulations of the pointest $\mathcal{C}$ (see end of Sect. 3.2). Then our method yields the actual implicit equation. If its parametrization is $x_0 = u\cos v,\ x_1 = u\sin v,\ x_2 = u^2$, which can be represented as rational functions

$$x_0 = \frac{t - ts^2}{1 + s^2},\ x_1 = \frac{2ts}{1 + s^2},\ x_2 = t^2,$$

then we define polynomials $F_0 = x_0(1 + s^2) - t + ts^2,\ F_1 = x_1(1 + s^2) - 2ts,\ F_2 = x_2 - t^2$ with supports $A_0 = \{(0,0),\ (0,2),\ (1,0),\ (1,2)\},\ A_1 = \{(0,0),\ (0,2),\ (1,1)\}$, and $A_2 = \{(0,0),\ (2,0)\}$. Their Cayley pointest is $\mathcal{C} = \{(0,0,0,0),\ (0,2,0,0),\ (1,0,0,0),\ (1,2,0,0),\ (0,0,1,0),\ (0,2,1,0),\ (1,1,1,0),\ (0,0,0,1),\ (2,0,0,1)\}$, and Q has vertices $(4,0,0), (0,4,0), (0,0,2)$. The matrix M has corank $= 4$ and all the polynomials corresponding to the kernel vectors are multiples of the implicit equation: $g_1 = (x_2 + x_0^2 - x_1^2)(-x_2 + x_0^2 + x_1^2),\ g_2 = x_0x_1(-x_2 + x_0^2 + x_1^2),\ g_3 = x_1^2(-x_2 + x_0^2 + x_1^2),\ g_4 = x_2(-x_2 + x_0^2 + x_1^2)$.

The following theorem establishes the relation between the dimension of the kernel of M and the accuracy of the predicted support. It remains valid even in the presence of base points. In fact, it also accounts for them, since then P is expected to be much smaller than Q.

Theorem 3.1 *Let $P = N(p)$ be the implicit polytope and Q the predicted polytope. Then, assuming M has been built using sufficiently generic evaluation points, the dimension of its kernel equals*

$$\#\{m \in \mathbb{Z}^n : m + P \subseteq Q\} = \#\{m \in \mathbb{Z}^n : N(x^m \cdot p) \subseteq Q\}.$$

Proof By Lemma 3.1, the kernel of M consists of the coefficient vectors $\vec{c}$ of all polynomials of the form fp, where $N(fp) \subset Q$, or, equivalently, $N(f) + N(p) \subset Q$.

Now, assume that there are precisely r elements $a_1, \dots, a_r \in \mathbb{Z}^n$ such that $N(x^{a_i} \cdot p) \subseteq Q$ and let $g_i = x^{a_i} p$, $i = 1, \dots, r$. Then the coefficient vector $\vec{c}_i$ of g_i lies in the kernel of M because g_i vanishes on all evaluation points $m_i(\tau_i)$, $i = 1, \dots, k$, used for constructing M, since p vanishes on these points. Moreover, the vectors $\vec{c}_i$ in the set $\{\vec{c}_1, \dots, \vec{c}_r\}$ are linearly independent. Every coefficient vector $\vec{c}$ of a polynomial of the form fp, where $N(fp) \subset Q$, can be written as a linear combination of the vectors $\vec{c}_i$, hence $\text{corank}(M) = r$. Since $P \subseteq Q$, it follows that $r \geq 1$, hence the nullspace of M has dimension ≥ 1.

Corollary 3.1 *Let M be the matrix from Algorithm 1, built with sufficiently generic evaluation points, and suppose the specialization of the corresponding polynomials to the parametric equations is sufficiently generic. Let $\{v_1, \dots, v_k\}$ be a basis of the kernel of M and $g_1, \dots, g_k$ be the corresponding polynomials (Step 4 of Algorithm 1). Then the GCD of $g_1, \dots, g_k$ equals the implicit equation.*

In practice the actual implicit equation can usually be found among the polynomials corresponding to the kernel vectors, or it can be obtained as the GCD of $\leq$ corank(M) (at least two) such polynomials, or via multivariate factoring of one polynomial corresponding to a kernel vector. Another approach to the problem is reduction of the predicted Newton polytope by discarding some of its vertices or by taking offsets. This allows to obtain a matrix of lesser corank or even of corank 1, and it can also be used for approximating the implicit equation.

3.3.2 Bernstein Basis

Our approach to implicitization relies on the support prediction method that operates in the monomial basis. However, current CAGD systems widely use parametric representations in the Bernstein basis where the data is often given as floating point numbers which prohibits exact computations.

To expand the applicability of our method, we experimented in approximating the implicit equation of such curves and surfaces. For practical purposes, such an approximation is expected to be of the lowest possible degree while reasonably close to the parametric curve or surface in the region of interest of the parameterization. In the process, the parameterization has to be converted to the monomial basis which may cause precision loss. To obtain a smaller predicted implicit polytope, we apply "filtering", i.e. we remove all monomials whose coefficients' absolute value is smaller than a threshold depending on the input by some educated guess (e.g. $<10^{-2}$). Finally, when building the matrix M we have to ensure that the evaluation points lie within the region of interest of the parameterization.

Table 3.1 Implicitization of Bézier and NURBS curves. Runtimes are given in seconds

Curve evaluation method	Degree	Exact solving rationals	SVD rationals	SVD Chebyshev nodes	SVD roots of unity
Bézier curve	4	0.16	0.12	0.13	0.21
Bézier curve	5	0.29	0.16	0.19	0.33
Bézier curve	6	0.47	0.16	0.21	0.53
Bézier curve	8	5.84	0.46	0.49	1.1
NURBS curve	3	–	0.05	0.05	0.09
NURBS curve	4	–	0.09	0.1	0.22
NURBS curve	7	–	0.43	0.36	0.93

Experiments performed on an Intel©Core2 Duo CPU, 2.20 GHz, 3Gb memory, `SAGE 5.4`

Table 3.1 contains Bézier curves of various degree which have integer coefficients when expressed in the monomial basis and NURBS curves from the package of industrial examples kindly provided by the authors of [24]. The NURBS curves have floating point coefficients, thus we are restricted to numerical methods.

In these experiments we have used evaluation by rational numbers, random or uniformly distributed, and complex roots of unity. Rational numbers provided both numerical stable and fast results. Note that for classical algebraic curves and surfaces evaluation by rational numbers led to a loss of numerical stability.

We have also tried evaluation with Chebyshev nodes in [0, 1]:

$$\tau = \frac{1}{2} + \frac{1}{2}\cos\left(\frac{2i-1}{2n}\pi\right),\ i = 1,\ \dots, n,$$

which allow to minimize the approximation error in numerical computations [1]. Complex roots of unity gave the slowest timings and introduced complex coefficients into the resulting approximate implicit equation.

3.3.3 Comparisons to Other Methods

We compare our `Maple` implementation against (a) `Maple`'s native function *Implicitize()* which employs integration over each parameter $t_1, \dots, t_n$, of the matrix obtained from $\vec{m}\vec{m}^\top$ after substitution of the variables x_i by the parametric expressions [5] (recall that $\vec{m}$ is a vector of monomials whose exponents form a superset of the support of the implicit equation), and (b) implicitization using Gröbner bases in `Maple`. Results are shown in Table 3.2. A highly effective linear solving function implemented in `Maple` (`LinearSolve`) results in exact computations being faster than the numerical ones.

Table 3.2 Comparison of our method (exact and numerical) to `Maple`'s function *Implicitize()* and Gröbner bases. Runtimes are given in seconds

Surface	Degree	LinearSolve	SVD	Implicitize()	Gröbner
Plücker's conoid	3	0.016	0.063	46.07	0.031
Plücker's conoid	5	0.016	0.063	85.43	0.046
Plücker's conoid	7	0.031	0.141	359.49	0.078
Plücker's conoid	9	0.046	0.202	695.65	0.078
Plücker's conoid	11	0.078	0.375	>2,000	0.141

Experiments performed on an Intel©Core2 Duo CPU, 2.20 GHz, 3 Gb memory, `Maple 14`

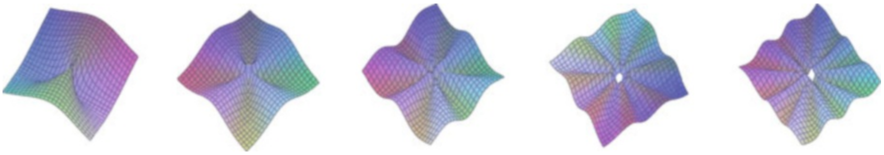


Fig. 3.1 Plücker's conoid surfaces used in Table 3.2. From the left to right: degree 3, 5, 7, 9, 11

The input consists of a family of classical algebraic surfaces, the so-called Plücker's conoid (Fig. 3.1): $x_0 = t,\ x_1 = s,\ x_2 = \frac{Re((t+i\cdot s)^a)}{|(t+i\cdot s)^a|}$, where i is the imaginary unit and $a \in \mathbb{N}$. The surfaces have a base point at $t = s = 0$. By choosing even values for the parameter a, we obtain rational parameterizations of the surfaces with the desired total degree.

The implicit polytope has been computed using `ResPol`. Timings shown in Table 3.2 concern only the interpolation part of the algorithm.

Of the three methods *Implicitize()* is the slowest, however the method has fewer restrictions on the parameterization, accepting non-rational representations. Our method is faster than *Implicitize()* and competitive to Gröbner bases.

This becomes more apparent as the degree and dimension grow: the ranunculoid curve (degree 12) was computed in 1.3 s by our method and in 7.3 s using Gröbner bases. For the standard benchmark of the bicubic surface (degree 18) the corresponding timings are 42 min and over 4 h, respectively.

Conclusion and Future Work

We have developed an algorithm for computing implicit equations that combines linear algebra with promising support prediction methods. The method applies to polynomial, rational and trigonometric parameterizations of classical algebraic equations of curves and (hyper)surfaces. Moreover, it can be used for implicitization of geometric objects represented in NURBS form, after converting them to the monomial basis. The method works even in the presence of base points.

(continued)

Its efficiency can be improved on the support prediction level, by taking offsets of the predicted polytope, which reduces the size of the matrix and its corank, and on the interpolation level, by taking advantage of the special structure of the constructed matrices. The method may also be used to attack the problem of implicitizing space curves, in which case the curve is defined as the intersection of the surfaces obtained by the kernel vectors. This will require the development of algorithms that compute the predicted implicit polytope from the Newton polytope of the Chow form of polynomials defined from the parameterization.

Acknowledgements Ioannis Z. Emiris and Tatjana Kalinka have been partially supported by Marie-Curie Initial Training Network "SAGA" (ShApes, Geometry, Algebra), FP7-PEOPLE contract PITN-GA-2008-214584. Most of the work was done while C. Konaxis was at Archimedes Center for Modeling, Analysis & Computation, University of Crete. C. Konaxis has received funding from the European Union's Seventh Framework Programme (FP7-REGPOT-2009-1) under grant agreement n^{o}245749. We thank Alicia Dickenstein for stating and proving Theorem 3.1, Thang Luu Ba for his contribution in evaluating the accuracy of the numerical method, and Oliver Barrowclough for his assistance with geometric objects in the Bernstein basis.

References

1. O. Barrowclough, T. Dokken, Approximate implicitization of triangular Bézier surfaces, in *Proceedings of the 26th Spring Conference on Computer Graphics, SCCG '10*, Budmerice, Slovakia (ACM, New York, 2010), pp. 133–140
2. B. Buchberger, Applications of Gröbner bases in non-linear computational geometry, in *Trends in Computer Algebra*, ed. by R. Janßen. Lecture Notes in Computer Science, vol. 296 (Springer, Berlin/Heidelberg, 1988), pp. 52–80. doi:10.1007/3-540-18928-9_5. http://dx.doi.org/10.1007/3-540-18928-9_5
3. L. Busé, Residual resultant over the projective plane and the implicitization problem, in *Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation*, London, Ontario, Canada (ACM, New York, 2001), pp. 48–55
4. E.W. Chionh, R.N. Goldman, Degree, multiplicity, and inversion formulas for rational surfaces using u-resultants. Comput. Aided Geom. Design **9**(2), 93–108 (1992)
5. R. Corless, M. Giesbrecht, I.S. Kotsireas, S. Watt, Numerical implicitization of parametric hypersurfaces with linear algebra, in *Proceeings of the AISC*, Madrid, pp. 174–183 (2000)
6. D. Cox, J. Little, D. O'Shea, *Ideals, Varieties, and Algorithms* (Springer, New York, 1996)
7. D. Cox, J. Little, D. O'Shea, *Using Algebraic Geometry*. Graduate Texts in Mathematics, vol. 185 (Springer, New York, 1998)
8. D. Cox, T. Sederberg, F. Chen, The moving line ideal basis of planar rational curves. Comput. Aided Geom. Des. **15**(8), 803–827 (1998)
9. C. D'Andrea, M. Sombra, The Newton polygon of a rational plane curve. Math. Comput. Sci. **4**(1), 3–24 (2010)
10. A. Dickenstein, E. Feichtner, B. Sturmfels, Tropical discriminants. J. AMS **20**, 1111–1133 (2007)
11. I.Z. Emiris, C. Konaxis, L. Palios, Computing the Newton polygon of the implicit equation. Math. Comput. Sci. Spec. Issue Comput. Geom. Comput. Aided Design **4**(1), 25–44 (2010)

12. I.Z. Emiris, I. Kotsireas, Implicit polynomial support optimized for sparseness, in *Proceedings of the International Conference on Computational Science and Its Application: Part III*, Montreal (Springer, Berlin, 2003), pp. 397–406. http://dl.acm.org/citation.cfm?id=1761839
13. I.Z. Emiris, V. Fisikopoulos, C. Konaxis, L. Peñaranda, An output-sensitive algorithm for computing projections of resultant polytopes. Int. J. Comput. Geom. Appl. **23**, 397–423 (2013). doi:10.1142/S0218195913600108
14. I.Z. Emiris, T. Kalinka, C. Konaxis, T. Luu Ba, Implicitization of curves and (hyper)surfaces using predicted support. Theor. Comput. Sci. Spec. Issue **479**(0), 81–98 (2013). doi:10.1016/j.tcs.2012.10.018. http://www.sciencedirect.com/science/article/pii/S0304397512009395
15. I.Z. Emiris, T. Kalinka, C. Konaxis, T. Luu Ba, Sparse implicitization by interpolation: characterizing non-exactness and an application to computing discriminants. Comput. Aided Des. Spec. Issue Symp. Phys. Model. 2013 **45**(2), 252–261 (2013)
16. A. Esterov, A. Khovanskii, Elimination theory and Newton polytopes. Funct. Anal. Other Math. **2**(1), 45–71 (2008). doi:10.1007/s11853-008-0015-2. http://dx.doi.org/10.1007/s11853-008-0015-2
17. V. Fisikopoulos, Respol: a software package for computing projections of resultant polytopes. Available from http://sourceforge.net/projects/respol
18. M. Franz, Convex: a Maple package for convex geometry, version 1.1.3. Available at: http://www-math.uwo.ca
19. X.S. Gao, S.C. Chou, Implicitization of rational parametric equations. J. Symb. Comput. **14**(5), 459–470 (1992). doi:http://dx.doi.org/10.1016/0747-7171(92)90017-X
20. I. Gelfand, M. Kapranov, A. Zelevinsky, Discriminants, resultants & multidimensional determinants (Birkhauser, Boston, 2008)
21. A. Jensen, J. Yu, Computing tropical resultants. J. Algebra **387**(0), 287–319 (2013). doi:http://dx.doi.org/10.1016/j.jalgebra.2013.03.031
22. D. Manocha, J.F. Canny, Algorithm for implicitizing rational parametric surfaces. Comput. Aided Geom. Des. **9**(1), 25–50 (1992). doi:10.1016/0167-8396(92)90051-P. http://dx.doi.org/10.1016/0167-8396(92)90051-P
23. T.W. Sederberg, F. Chen, Implicitization using moving curves and surfaces, in *Annual Conference on Computer Graphics*, Los Angeles, CA, USA, pp. 301–308 (1995). doi:10.1145/218380.218460
24. M. Shalaby, J. Thomassen, E. Wurm, T. Dokken, B. Jüttler, Piecewise approximate implicitization: experiments using industrial data, in *Algebraic Geometry and Geometric Modeling*, ed. by B. Mourrain, M. Elkadi, R. Piene (Springer, Berlin/Heidelberg, 2006), pp. 37–51
25. B. Sturmfels, On the Newton polytope of the resultant. J. Algebr. Comb. **3**, 207–236 (1994)
26. B. Sturmfels, J. Tevelev, J. Yu, The Newton polytope of the implicit equation. Mosc. Math. J. **7**(2), 327–346 (2007)
27. B. Sturmfels, J. Yu, Minimal polynomials and sparse resultants, in *Proceedings of the Zero-Dimensional Schemes*, ed. by F. Orecchia, L. Chiantini, Ravello, 1992 (De Gruyter, 1994), pp. 317–324
28. B. Sturmfels, J. Yu, Tropical implicitization and mixed fiber polytopes, in *Software for Algebraic Geometry*. IMA Volumes in Mathematical & Its Applications, vol. 148 (Springer, New York, 2008), pp. 111–131

Chapter 4
The Intersection Problems of Parametric Curves and Surfaces by Means of Matrix-Based Implicit Representations

Thang Luu Ba

4.1 Introduction

Rational algebraic curves and surfaces can be described in different ways, the most common being the parametric and implicit representations. Parametric representations describe the geometric object as the closed image of a rational map and implicit representations describe it as the zero set of polynomial equations. Both representations have a wide range of applications in Computer Aided Geometric Design and Geometric Modeling. A parametric representation is much convenient for drawing a surface but less appropriate for checking if a point lies on a surface whereas the converse holds for the implicit representation.

The matrix-based implicit representation of parametric curves and parametric surfaces has been addressed many times in literature (for example [6, 11, 21, 23]). However, it has usually been by writing the implicit equation as the determinant of a square matrix. The case of planar curves is particularly well known because one always knows how to find such a simple square matrix. One can read the article of T. W. Sederberg and S.R. Parry [24] who seemed to introduce this technique to the problems intersection between plane curves for modeling geometric. The case of parametric surfaces is especially much more difficult because the geometry of their parameterizations becomes richer with the inevitable appearance of base points (these are points where a parameterization is not well defined). In order to find a square matrix whose determinant is an implicit equation, one must be restricted to particular classes of parameterizations [6, 11, 16], which turns out to be very restrictive in practice.

T. Luu Ba (✉)
Department of Mathematics and Informatics, Hanoi National University of Education, 136 Xuan Thuy, Cau Giay, Hanoi, Vietnam

INRIA, Galaad, 2004 route des Lucioles, 06902 Sophia Antipolis, France
e-mail: thanglb@hnue.edu.vn

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_4

In this paper, we show how, by releasing the constraint matrix square, we can easily form an implicit matrix-based representation for a very general class of parametric surfaces. The matrix in question is no longer square, but still allows to characterize the surface: the cancellation of a determinant is here replaced here by a drop in rank. In addition, treatment of intersection problems can be reduced to linear algebra computations, allowing the use of robust tools and approximate calculation, such as the singular value decomposition, calculating eigenvalues, and generalized eigenvectors. Note that these implicit representation matrices can be seen as a bridge between the parametric representation of a curve, surface and its implicit representation.

This article covers a series of works [3–5, 7, 8, 10], which led to the notion of the implicit representation matrix of a parametric curve or a parametric surface, together with the development of applications for intersection problems in geometric modeling [8, 9, 20]. Its content is part of the authour's PhD thesis [19].

4.2 Matrix Based Implicit Representations of Parametric Surfaces

4.2.1 Construction of Matrix Representations

Given a parametric rational surface, we briefly recall from [3, 10] how to build a matrix that *represents* this surface in a sense that we will make explicit. So suppose given a parameterization

$$\mathbb{P}^2_{\mathbb{R}} \xrightarrow{\phi} \mathbb{P}^3_{\mathbb{R}}$$
$$(s:t:u) \mapsto (f_1 : f_2 : f_3 : f_4)(s,t,u)$$

of a surface $\mathbf{S}$ such that $\gcd(f_1,\dots,f_4) \in \mathbb{R} \setminus \{0\}$. Set $d := \deg(f_i) \geq 1$, $i = 1,2,3,4$, and denote by x, y, z, w the homogeneous coordinates of the projective space $\mathbb{P}^3_{\mathbb{R}}$. Notice that s, t, u are the homogeneous parameters of the surface $\mathbf{S}$ and that an affine parameterization of $\mathbf{S}$ can be obtained by "inverting" one of these parameters; for instance, setting $s' = s/u$ and $t' = t/u$ we get the following affine parameterization of $\mathbf{S}$:

$$\mathbb{R}^2 \xrightarrow{\phi} \mathbb{R}^3$$
$$(s',t') \mapsto \left(\frac{f_1(s',t',1)}{f_4(s',t',1)}, \frac{f_2(s',t',1)}{f_4(s',t',1)}, \frac{f_3(s',t',1)}{f_4(s',t',1)}\right)$$

The implicit equation of $\mathbf{S}$ is a homogeneous polynomial $S(x,y,z,w) \in \mathbb{R}[x,y,z,w]$ of smallest degree such that $S(f_1, f_2, f_3, f_4) = 0$ (observe that it is defined up to multiplication by a nonzero element in $\mathbb{R}$). It is well known that the quantity $\deg(\mathbf{S})\deg(\phi)$ is equal to d^2 minus the number of common roots of f_1, f_2, f_3, f_4 in $\mathbb{P}^2_{\mathbb{R}}$, that are called *base points* of the parameterization ϕ, counted with suitable multiplicities (see for instance [10, Theorem 2.5] for more details). The notation deg(**S**) stands for the degree of the surface **S**, which is nothing but the degree of the implicit equation of **S**.

The notation deg(ϕ) stands for the degree of the parameterization ϕ (co-restricted to **S**) that, roughly speaking, measures the number of times the surface **S** is drawn by the parameterization ϕ. More precisely, deg(ϕ) is equal to the number of pre-images of a general point on **S** by the parameterization ϕ.

For every non-negative integer ν, we build a matrix $\mathtt{M}(\phi)_\nu$ as follows. Consider the set $\mathcal{L}(\phi)_\nu$ of polynomials of the form

$$a_1(s,t,u)x + a_2(s,t,u)y + a_3(s,t,u)z + a_4(s,t,u)w$$

such that

- $a_i(s,t,u) \in \mathbb{R}[s,t,u]$ is homogeneous of degree ν for $i = 1,\dots,4$,
- $\sum_{i=1}^4 a_i(s,t,u) f_i(s,t,u) \equiv 0$ in $\mathbb{R}[s,t,u]$.

The set $\mathcal{L}(\phi)_\nu$ has a natural structure of $\mathbb{R}$-vector space of finite dimension because each polynomial $a_i(s,t,u)$ is homogeneous of degree ν and that the set of homogeneous polynomials of degree ν in the variables s,t,u is an $\mathbb{R}$-vector space of dimension $\binom{\nu+2}{2}$ with canonical basis the set of monomials $\{s^\nu, s^{\nu-1}t, \cdots, u^\nu\}$. So, denote by $L^{(1)},\dots,L^{(n_\nu)}$ a basis of the $\mathbb{R}$-vector space $\mathcal{L}(\phi)_\nu$; it can be computed by solving a single linear system whose indeterminates are the coefficients of the polynomials $a_i(s,t,u)$, $i = 1,2,3,4$. The matrix $\mathtt{M}(\phi)_\nu$ is then by definition the matrix of coefficients of $L^{(1)},\dots,L^{(n_\nu)}$ as homogeneous polynomials of degree ν in the variables s,t,u. In other words, we have the equality of matrices:

$$\left[s^\nu \; s^{\nu-1}t \cdots u^\nu\right] \mathtt{M}(\phi)_\nu = \left[L^{(1)} \; L^{(2)} \cdots L^{(n_\nu)}\right].$$

Notice that we have chosen for simplicity the monomial basis for the $\mathbb{R}$-vector space of homogeneous polynomials of degree ν in s,t,u. However, any other choice, for instance the Bernstein basis, can be made without affecting the result.

For every integer $\nu \geq 2d-2$, the matrix $\mathtt{M}(\phi)_\nu$ is said to be a *representation matrix* of ϕ because it satisfies the following properties under the assumption that the base points of ϕ, if any, form *locally a complete intersection*, which means that at each base point, the ideal of polynomials (f_1, f_2, f_3, f_4) can be generated by two equations (see [10, Definition 4.8] for more details):

- The entries of $\mathtt{M}(\phi)_\nu$ are linear forms in $\mathbb{R}[x,y,z,w]$.
- The matrix $\mathtt{M}(\phi)_\nu$ has $\binom{\nu+2}{2}$ rows (which is nothing but the dimension of the $\mathbb{R}$-vector space of homogeneous polynomials of degree ν in three variables, here s,t,u) and possesses at least as many columns as rows.
- The rank of $\mathtt{M}(\phi)_\nu$ is $\binom{\nu+2}{2}$(the rank of $\mathtt{M}(\phi)_\nu$ measures the independency of the columns (and the rows) as linear combinations with coefficients in $\mathbb{R}$).
- When specializing $\mathtt{M}(\phi)_\nu$ at a given point $P \in \mathbb{P}^3_{\mathbb{R}}$, its rank drops if and only if P belongs to **S**.

- The greatest common divisor of the $\binom{\nu+2}{2}$-minors of $\mathtt{M}(\phi)_\nu$ is equal to the implicit equation of **S** raised to the power $\deg(\phi)$.

From a computational point of view, the matrix $\mathtt{M}(\phi)_\nu$ with the smallest possible value of ν has to be chosen. It is rarely a square matrix. Also, notice that the last property given above is never used for computations; our aim is to keep the matrix $\mathtt{M}(\phi)_\nu$ as an implicit representation of **S** in place of its implicit equation.

There are many results that lead to enlarge the above family of matrices and to make it available in other contexts. Since a detailed overview of these results is not the main purpose of this paper, we just recall them shortly with appropriate references to the literature:

- The hypothesis on the base points of ϕ can be relaxed. If the base points are locally *almost* a complete intersection, meaning that they are locally given by three (and not two) equations, then the above family of matrices can still be constructed and provide a matrix representation of the surface **S** plus a certain product of hyperplanes that can be described from the parameterization ϕ. In addition, the lower bound $2d-2$ for the integer ν can be decreased. See [3, 10].
- In our setting, ϕ parameterizes what is called a triangular Bézier patch. It turns out that a similar family of matrices $\mathtt{M}(\phi)_\nu$ can be built for parameterizations of tensor product surfaces, and even for any parameterization whose parameter space is a projective toric variety (triangular and tensor product surfaces are particular cases of parameterizations whose parameter space is a projective toric variety). We refer the interested reader to [2, 7].
- To build the matrices $\mathtt{M}(\phi)_\nu$ we used what is called moving planes, that is to say syzygies of the parameterization ϕ. It is actually possible to build another family of matrices by taking into account moving quadrics, i.e. syzygies associated to the square of the ideal generated by the parameterization of ϕ. In this way, we get a family containing smaller matrices whose entries are either linear or quadratic forms in $\mathbb{R}[x, y, z, w]$. In some sense, they generalize the matrices given in [11] and [6]. See [4].

Example 4.1 The Steiner surface **S** of degree 2 parameterized by

$$\phi_1 : \mathbb{P}^2 \to \mathbb{P}^3 : (s : t : u) \mapsto (s^2 + t^2 + u^2 : tu : st : su)$$

admits the matrix representation

$$M(x, y, z, w) := \begin{pmatrix} -x & 0 & -y & 0 & -y & y & 0 & z & 0 \\ y & -y & 0 & w & 0 & -x & -y & 0 & 0 \\ 0 & 0 & w & 0 & 0 & 0 & z & 0 & -x \\ w & 0 & 0 & -y & 0 & z & 0 & -y & y \\ 0 & w & 0 & 0 & 0 & z & 0 & 0 & y \\ w & 0 & 0 & 0 & z & 0 & 0 & 0 & y \end{pmatrix}.$$

Example 4.2 Let **S** be the rational surface of degree 3 that is parametrized by

$$\phi : \mathbb{P}^2 \to \mathbb{P}^3 : (s : t : u) \mapsto (f_1 : f_2 : f_3 : f_4)$$

where

$$f_1 = s^3 + t^2u,\, f_2 = s^2t + t^2u,\, f_3 = s^3 + t^3,\, f_4 = s^2u + t^2u.$$

Then, a matrix representation of **S** is

$$\begin{pmatrix} 0 & 0 & 0 & w-y & 0 & 0 & z-x \\ w & 0 & 0 & x & w-y & 0 & 0 \\ x-y-z & 0 & 0 & -z & 0 & w-y & 0 \\ 0 & w & 0 & 0 & x & 0 & -y \\ 0 & x-y-z & w & 0 & -z & x & y+z-x \\ 0 & 0 & x-y-z & 0 & 0 & -z & 0 \end{pmatrix}$$

4.2.2 Points on Surface and Inversion Problem

Suppose given a parameterization ϕ of a parametric surface **S** and a point P in $\mathbb{P}^3$. Denote by $\mathtt{M}(\phi)_\nu$ a matrix representation of ϕ for some integer $\nu \geq 2d - 2$. Since its entries are linear forms in the variables x, y, z, w, one can evaluate $\mathtt{M}(\phi)_\nu$ at P and get a matrix with coefficients in the ground field $\mathbb{R}$. Then, we have that

$$\operatorname{rank}\left(\mathtt{M}(\phi)_\nu(P)\right) < \binom{\nu+2}{2} \text{ if and only if } P \in \mathbf{S}.$$

This property answers the point-on curve problem.

If $\operatorname{rank}\mathtt{M}(\phi)_\nu(P) = \operatorname{rank}\mathtt{M}(\phi)_\nu - 1 = \binom{\nu+2}{2} - 1$ then P has a unique pre-image $(s_0 : t_0 : u_0)$ by ϕ and moreover, this pre-image can be recovered from the computation of a generator, say $W_P = (w_0, \ldots, w_{\binom{\nu+2}{2}-1}) \in \mathbb{R}^{\binom{\nu+2}{2}}$, of the kernel of the transpose of $\mathtt{M}(\phi)_\nu(P)$. Indeed, if $b_0(s,t,u), \ldots, b_{\binom{\nu+2}{2}-1}(s,t,u)$ is the basis of $\mathbb{R}[s,t,u]_\nu$ that has been chosen to build $\mathtt{M}(\phi)_\nu$, then there exists $\lambda \in \mathbb{R} \setminus \{0\}$ such that

$$W_P = \lambda\left(b_0(s_0,t_0,u_0), \ldots, b_{\binom{\nu+2}{2}-1}(s_0,t_0,u_0)\right).$$

For instance, suppose that $(b_0(s,t,u), \ldots, b_{\binom{\nu+2}{2}-1}(s,t,u)) = (s^i t^j u^{\nu-i-j},\, 0 \leq i,j \leq \nu, i+j \leq \nu)$ (the usual monomial basis), then $(s_0 : t_0 : u_0) = (w_2 : w_1 : w_0)$.

Obviously, the inversion problems have been translated into compute kernel of the transpose of $\mathtt{M}(\phi)_\nu(P)$, for which there exits numerically effective algorithms like the Singular Value Decomposition.

We also point out that the points $P \in \mathbf{S}$ such that $\operatorname{rank}\mathtt{M}(\phi)_\nu(P) = \operatorname{rank}\mathtt{M}(\phi)_\nu - 1 = \binom{\nu+2}{2} - 1$ are precisely the regular points on $\mathbf{S}$. However, the conversion doesn't hold.

4.3 Curve/Surface Intersection

Suppose given a parametric surface $\mathbf{S}$ represented by a homogeneous and irreducible implicit equation $S(x, y, z, w) = 0$ in $\mathbb{P}^3_{\mathbb{R}}$ and a rational space curve $\mathcal{C}$ represented by a parameterization

$$\phi : \mathbb{P}^1_{\mathbb{R}} \to \mathbb{P}^3_{\mathbb{R}} : (s : t) \mapsto (x(s,t) : y(s,t) : z(s,t) : w(s,t)) \tag{4.1}$$

where $x(s,t), y(s,t), z(s,t), w(s,t)$ are homogeneous polynomials of the same degree and without common factor in $\mathbb{R}[s,t]$.

A standard problem in nonlinear computational geometry is to determine the set $\mathcal{C} \cap \mathbf{S} \subset \mathbb{P}^3_{\mathbb{R}}$, especially when it is finite. One way to proceed, is to compute the roots of the homogeneous polynomial

$$S(x(s,t), y(s,t), z(s,t), w(s,t)) \tag{4.2}$$

because they are in correspondence with $\mathcal{C} \cap \mathbf{S}$ through the regular map ϕ. Observe that (4.2) is identically zero if and only if $\mathcal{C} \cap \mathbf{S}$ is infinite, equivalently $\mathcal{C} \subset \mathbf{S}$ (for $\mathcal{C}$ is irreducible).

Assume that $M(x, y, z, w)$ is a matrix representation of the surface $\mathbf{S}$, meaning a representation of the polynomial $S(x, y, z, w)$. By replacing the variables x, y, z, w by the homogeneous polynomials $x(s,t), y(s,t), z(s,t), w(s,t)$ respectively, we get the matrix

$$M(s,t) = M(x(s,t), y(s,t), z(s,t), w(s,t)).$$

Therefore, we have the following easy property: for every point $(s_0 : t_0) \in \mathbb{P}^1_{\mathbb{R}}$ the rank of the matrix $M(s_0, t_0)$ drops if and only if the point $(x(s_0,t_0) : y(s_0,t_0) : z(s_0,t_0) : w(s_0,t_0))$ belongs to the intersection locus $\mathcal{C} \cap \mathbf{S}$.

It follows that points in $\mathcal{C} \cap \mathbf{S}$ associated to points $(s : t)$ such that $s \neq 0$, are in correspondence with the set of values $t \in \mathbb{R}$ such that $M(1,t)$ drops of rank strictly less than its row and column dimensions, i.e., the set of generalized eigenvalues of $M(1,t)$ that this will be explained in detail in the next section. Now, we present a technique from linear algebra which allows us to obtain the regular part of the pencil matrices.

4.3.1 Linearization of a Polynomial Matrix in the Monomial Basis

We begin with some notation. Let A and B be two matrices of size $m \times n$ with coefficients in $\mathbb{R}$. We will call a *generalized eigenvalue* of A and B a value in the set

$$\lambda(A, B) := \{t \in \mathbb{R} : \operatorname{rank}(A - tB) < \min\{m, n\}\}.$$

In the case $m = n$, the matrices A and B have n generalized eigenvalues if and only if $\operatorname{rank}(B) = n$. If $\operatorname{rank}(B) < n$, then $\lambda(A, B)$ can be finite, empty or infinite. Moreover, if B is invertible then $\lambda(A, B) = \lambda(AB^{-1}, I) = \lambda(AB^{-1})$, which is the ordinary spectrum of AB^{-1}.

Suppose given an $m \times n$-matrix $M(t) = (a_{i,j}(t))$ with polynomial entries $a_{i,j}(t) \in \mathbb{R}[t]$. It can be equivalently written as a polynomial in t with coefficients $m \times n$-matrices with entries in $\mathbb{R}$: if $d = \max_{i,j}\{\deg(a_{i,j}(t))\}$ then

$$M(t) = M_d t^d + M_{d-1} t^{d-1} + \ldots + M_0$$

where $M_i \in \mathbb{R}^{m \times n}$.

The generalized companion matrices A, B of the matrix $M(t)$ are the matrices with coefficients in $\mathbb{R}$ of size $((d-1)m + n) \times dm$ that are given by

$$A = \begin{pmatrix} 0 & I_m & \ldots\ldots & & 0 \\ 0 & 0 & I_m & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & I_m \\ M_0^t & M_1^t & \ldots\ldots & & M_{d-1}^t \end{pmatrix}, B = \begin{pmatrix} I_m & 0 & \ldots\ldots & & 0 \\ 0 & I_m & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & I_m & 0 \\ 0 & 0 & \ldots & 0 & -M_d^t \end{pmatrix}$$

where I_m stands for the identity matrix of size m and M_i^t stands for the transpose of the matrix M_i. These companion matrices allow to *linearize* the polynomial matrix $M(t)$ in the sense that there exists two unimodular matrices $E(t)$ et $F(t)$, i.e., invertible matrices with non-vanishing determinant independent of t, with entries in $\mathbb{R}[t]$ and of size dm and $(d-1)m + n$ respectively, such that

$$E(t)\,(A - tB)\,F(t) = \left(\begin{array}{c|c} {}^{t}M(t) & 0 \\ \hline 0 & I_{m(d-1)} \end{array}\right). \qquad (4.3)$$

Then, we have

$$\operatorname{rank} M(t) \text{ drops} \Leftrightarrow \operatorname{rank}(A - tB) \text{ drops}.$$

We also refer the reader to [14, 20] for more details. We call t such that $\operatorname{rank}(A - tB)$ drops the generalized eigenvalues of the pencil of matrices $A - tB$. We thus transformed the computation of generalized eigenvalues of the matrix polynomial $M(t)$ into the computation of generalized eigenvalues of a pencil of matrices $A - tB$. If the matrices A, B were two square matrices, then we could compute their generalized eigenvalues by the QZ-algorithm [15]. Therefore, our next task is to reduce the pencil $A - tB$ into a square pencil that keeps the information we are interested in.

4.3.2 Extracting the Regular Part of a Non-square Pencil of Matrices

For any couple constant matrices A, B of size $p \times q$ such that $A - tB$ is full rank, there exist constant invertible matrices P and Q such that the pencil $P(A - tB)Q$ is of the block-diagonal form

$$\operatorname{diag}\{L_{i_1}, \dots, L_{i_s}, L^t_{j_1}, \dots, L^t_{j_u}, \Omega_{k_1}, \dots, \Omega_{k_v}, A' - tB'\}$$

where A', B' are square matrices, B' is invertible and $L^t_{j_l}$ is the transpose of L_{j_l} for $l = 1, \dots, u$. The dimensions $i_1, \dots, i_s, j_1, \dots, j_u, k_1, \dots, k_v$ and the determinant of $A' - tB'$ (up to a scalar) are independent of the representation. Here $L_k(t)$, $\Omega_k(t)$ are the two matrices of size $k \times (k + 1)$ and $k \times k$ respectively, defined by

$$L_k(t) = \begin{pmatrix} 1 & t & 0 & \dots & 0 \\ 0 & 1 & t & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 1 & t & 0 \\ 0 & 0 & \dots & 1 & t \end{pmatrix}, \Omega_k(t) = \begin{pmatrix} 1 & t & 0 & \dots & 0 \\ 0 & 1 & t & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 1 & t \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}.$$

This form is called the *Kronecker canonical form* of a pencil of matrices (see for instance [13, p. 35–37]). Notice that if the pencil $A - tB$ is not full rank then there is a zero matrix in Kronecker canonical form.

It is interesting to notice that the above decomposition can be computed within $O(p^2q)$ arithmetic operations. We refer the reader to [1] for a proof, as well as for an analysis of the stability of this decomposition.

Following the ideas developed in [1] and the reduction methods exploited in [22], we now describe an algorithm that allows to remove the Kronecker blocks L_k, L^t_k and Ω_k from the pencil of matrices $A - tB$ in order to extract the regular pencil $A' - tB'$. We also refer the reader to [20] for more details.

We start with a pencil $A - tB$ where A, B are constant matrices of size $p \times q$ with coefficients in a field $\mathbb{R}$. Set $\rho = \operatorname{rank} B$. In the following algorithm, all computational steps are realized via the classical LU-decomposition.

- Transform B into its column echelon form; that amounts to determining unitary matrices P_0 and Q_0 such that

$$B_1 = P_0 B Q_0 = [\underbrace{B_{1,1}}_{\rho} \mid \underbrace{0}_{q-\rho}]$$

  where $B_{1,1}$ is an echelon matrix. Then, compute

$$A_1 = P_0 A Q_0 = [\underbrace{A_{1,1}}_{\rho} \mid \underbrace{A_{1,2}}_{q-\rho}]$$

- Transform $A_{1,2}$ into its row echelon form; that amounts to determine unitary matrices P_1 and Q_1 such that

$$P_1 A_{1,2} Q_1 = \left(\begin{array}{c} A'_{1,2} \\ \hline 0 \end{array}\right)$$

  where $A'_{1,2}$ has full row rank while keeping $B_{1,1}$ in echelon form.
  Put $Q'_1 = \left(\begin{array}{c|c} I_\rho & 0 \\ \hline 0 & Q_1 \end{array}\right)$, I_ρ is the identity matrix of size ρ. Then,

$$P_1 A_1 Q'_1 = \left(\begin{array}{c|c} A'_{1,1} & A'_{1,2} \\ \hline A_2 & 0 \end{array}\right); P_1 B_1 Q'_1 = \left(\begin{array}{c|c} B'_{1,1} & 0 \\ \hline B_2 & 0 \end{array}\right)$$

  Thus we obtain a new pencil of matrices, namely $A_2 - tB_2$.
- Starting from $j = 2$, repeat the above steps 1 and 2 for the pencil $A_j - tB_j$ until the $p_j \times q_j$ matrix B_j has full column rank, that is to say until rank $B_j = q_j$.
- If B_j is not a square matrix, then we repeat the above procedure with the transposed pencil $A^t_j - tB^t_j$.

At last, we obtain the regular pencil $A' - tB'$ where A', B' are two square matrices and B' is invertible. Moreover, we have the

$$\operatorname{rank}(A - tB) \text{ drops} \Leftrightarrow \operatorname{rank}(A' - tB') \text{ drops}.$$

We are now ready to state our algorithm for solving the curve/surface intersection problem:

Algorithm 1: Matrix intersection algorithm

```
Input: A matrix representation of a surface S and a parametrization (4.1) of a
       rational space curve $\mathcal{C}$ not contained in S.
Output: The intersection points of S and $\mathcal{C}$.
1. Compute the matrix representation M(t).
2. Compute the generalized companion matrices A and B of M(t).
3. Compute the companion regular matrices $A'$ and $B'$.
4. Compute the generalized eigenvalues of $(A', B')$.
5. For each eigenvalue $t_0$, the point
$\phi(1:t_0) = P(x(1,t_0) : y(1,t_0) : z(1,t_0) : w(1,t_0))$ is one of the intersection
points.
```

Remark that this algorithm returns all the points in $\mathcal{C} \cap \mathbf{S}$ except possibly the point $\phi(1:0)$. This point can be treated independently.

Example 4.3 Let **S** be the sphere that we suppose given as the image of the parametrization

$$\phi : \mathbb{P}^2 \to \mathbb{P}^3 : (s:t:u) \mapsto (f_1 : f_2 : f_3 : f_4)$$

where

$$f_1 = s^2 + t^2 + u^2,\ f_2 = 2su,\ f_3 = 2st,\ f_4 = s^2 - t^2 - u^2.$$

Let $\mathcal{C}$ be the twisted cubic which is parametrized by

$$x(t) = 1, y(t) = t, z(t) = t^2, w(t) = t^3.$$

The computation of a matrix representation of the sphere **S** gives

$$\begin{pmatrix} -y & 0 & z & x+w \\ 0 & -y & -x+w & -z \\ z & x+w & y & 0 \end{pmatrix}.$$

Now, a point P belongs to the intersection of **S** and $\mathcal{C}$ if and only if $P = (1 : t : t^2 : t^3)$ and t is one of the generalized eigenvalues of the matrix

$$M(t) = \begin{pmatrix} -t & 0 & t^2 & 1+t^3 \\ 0 & -t & -1+t^3 & -t^2 \\ t^2 & 1+t^3 & t & 0 \end{pmatrix}.$$

As before, we compute the generalized eigenvalues and find:

$$t_1 \approx 0.7373527056, t_2 \approx -0.7373527056,$$

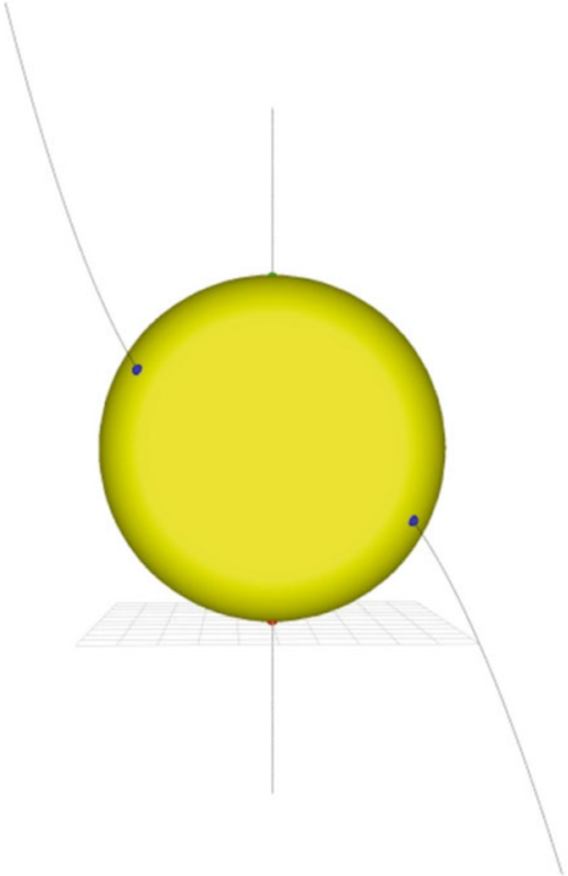


Fig. 4.1 Intersection of the sphere and the twisted cubic, the axis Oz

$$t_3 \approx 0.5405361044 + 1.031515287i, t_4 \approx -0.5405361044 - 1.031515287i,$$

$$t_5 \approx 0.5405361044 - 1.031515287i, t_6 \approx -0.5405361044 + 1.031515287i.$$

All these eigenvalues have multiplicity 1. They all correspond to one intersection point of $\mathbf{S}$ and $\mathcal{C}$ which has multiplicity 1. By Bézout Theorem, there are all the intersection points between these two algebraic varieties (all of them are at finite distance) (Fig. 4.1).

4.4 Surface/Surface Intersection

Computing the intersection between two parametric algebraic surfaces is a fundamental task in Computer Aided Geometric Design. Several methods and approaches have been developed for that purpose. Some of them are based on the use of matrix representations of the objects because they allow to transform geometric operations on the intersection curve into matrix operations. This approach seems to have been first introduced by J. Canny and D. Manocha in their paper [21]. Roughly speaking, it amounts to representing the implicit equation of one of the two surfaces as the determinant of a certain matrix, necessarily square. Then, instead of using this implicit equation, the matrix itself is used as a representation of this first parametric surface and then a matrix representation of the intersection curve is easily obtained by replacing the implicit variables by the parameterization of the second surface. In this section, we extend the approach of Canny and Manocha concerning surface/surface intersection to the significantly larger class of parameterizations introduced in Sect. 4.2.

Suppose given two distinct parametric surfaces $\mathbf{S}_1$ and $\mathbf{S}_2$. A standard problem in nonlinear computational geometry is to determine the set $\mathbf{S}_1 \cap \mathbf{S}_2$ which is a curve in $\mathbb{P}^3_{\mathbb{R}}$. As we explained above, one can build a representation matrix of $\mathbf{S}_1$ that we will denote by $M(x, y, z, w)$. Let

$$\phi : \mathbb{P}^2_{\mathbb{R}} \to \mathbb{P}^3_{\mathbb{R}} : (s : t : u) \mapsto (a(s,t,u) : b(s,t,u) : c(s,t,u) : d(s,t,u))$$

be a parameterization of $\mathbf{S}_2$ where $a(s,t,u), b(s,t,u), c(s,t,u), d(s,t,u)$ are homogeneous polynomials of the same degree and without common factor in $\mathbb{R}[s,t,u]$. By substituting in the matrix $M(x, y, z, w)$ the variables x, y, z, w by the homogeneous polynomials $a(s,t,u), b(s,t,u), c(s,t,u), d(s,t,u)$ respectively, we get the matrix

$$M(s,t,u) := M(\phi(s : t : u)) = M(a(s,t,u), b(s,t,u), c(s,t,u), d(s,t,u)).$$

From the properties of the representation matrix $M(x, y, z, w)$, we know that $M(s,t,u)$ has maximal rank ρ (where ρ is the number of rows of M). Moreover, for every point $(s_0 : t_0 : u_0) \in \mathbb{P}^2_{\mathbb{R}}$ we have

$$\operatorname{rank}(M(s_0, t_0, u_0)) < \rho \text{ if and only if } \begin{cases} \phi(s_0 : t_0 : u_0) \in \mathbf{S}_1 \cap \mathbf{S}_2 \text{ or} \\ (s_0 : t_0 : u_0) \;\text{ is a base point of }\; \phi. \end{cases} \tag{4.4}$$

The equivalence (4.4) shows that the spectrum of the matrix $M(s,t,u)$, that is to say the set

$$\left\{(s_0 : t_0 : u_0) \in \mathbb{P}^2_{\mathbb{R}} \text{ such that } \operatorname{rank} M(s_0, t_0, u_0) < \rho\right\},$$

yields the intersection locus $\mathbf{S}_1 \cap \mathbf{S}_2$ plus the base points of the parameterization ϕ of $\mathbf{S}_2$.

In [9] we proved that the spectrum of the matrix $M(s,t,u)$ is an algebraic curve in $\mathbb{P}^2_{\mathbb{R}}$, that is to say is equal to the zero locus of a homogeneous polynomial in $\mathbb{R}[s,t,u]$. In particular, there are no isolated points in the spectrum of $M(s,t,u)$. As a consequence if we use matrix representations to deal with the surface/surface intersection problem, we will at some point end up with a pencil of bivariate and non-square matrices that represents the intersection curve (after dehomogenization). Therefore, in order to be able to handle this intersection curve, for instance to determine its exact topology, it is necessary to extract a pencil of bivariate and square matrices that yields a matrix representation of the intersection curve as a matrix determinant. For that purpose, we develop an algorithm (called ΔW-Decomposition) based on the remarkable work of V. N. Kublanovskaya [17, 18].

We build two companion matrices $A(t)$ and $B(t)$ which allow us to *linearize* the polynomial matrix $M(s,t,1)$ such that the spectrum of the matrix $M(s,t,1)$ coincides the spectrum of the matrix $A(t) - sB(t)$. Then, we provide an algorithm that extracts a square matrix whose determinant represents the intersection locus $\mathbf{S}_1 \cap \mathbf{S}_2$. A pencil of polynomial matrices $A(t) - sB(t)$ is equivalent to a pencil of the following form

$$P(t)(A(t) - sB(t))Q(t) = \begin{pmatrix} M_{1,1}(s,t) & 0 & 0 \\ M_{2,1}(s,t) & M_{2,2}(s,t) & 0 \\ M_{3,1}(s,t) & M_{3,2}(s,t) & M_{3,3}(s,t) \end{pmatrix}$$

where $P(t), Q(t)$ are unimodular matrices and the pencil $M_{2,2}(s,t)$ is a regular pencil that corresponds to the intersection locus $\mathbf{S}_1 \cap \mathbf{S}_2$.

Now, we get the following algorithm (for more details see [9]):

Algorithm 2: Matrix representation of an intersection curve

```
Input: Two parametric algebraic surfaces $S_1$ and $S_2$ such that the parameterization of $S_1$ has
       local complete intersection base points.
Output: The intersection curve $S_1 \cap S_2$ represented as a matrix determinant.
1. Compute a matrix representation of $S_1$, say $M(x, y, z, w)$.
2. Replace $x, y, z, w$ by the parameterization of $S_2$ in the matrix $M$ to get a matrix $M(s, t)$
(set $u = 1$).
3. Compute the generalized companion matrices $A(s)$ and $B(s)$ associated to $M(s, t)$.
4. Return the regular pencil of matrices $M_1(s, t) = A_1(s) - tB_1(s)$.
```

In comparison with [21], our algorithm returns a result of the same type: a determinant matrix representation of the intersection curve, but the class of parameterizations of surfaces for which step 1 can be performed is here dramatically extended. We present an illustrative example.

Example 4.4 We start with the Steiner surface S_1 parameterized by

$$\phi_1 : \mathbb{P}^2 \to \mathbb{P}^3 : (s : t : u) \mapsto (s^2 + t^2 + u^2 : tu : st : su)$$

which admits the matrix representation

$$M(x,y,z,w) := \begin{pmatrix} -x & 0 & -y & 0 & -y & y & 0 & z & 0 \\ y & -y & 0 & w & 0 & -x & -y & 0 & 0 \\ 0 & 0 & w & 0 & 0 & 0 & z & 0 & -x \\ w & 0 & 0 & -y & 0 & z & 0 & -y & y \\ 0 & w & 0 & 0 & 0 & z & 0 & 0 & y \\ w & 0 & 0 & 0 & z & 0 & 0 & 0 & y \end{pmatrix}.$$

We want to study the intersection between S_1 and the cubic surface S_2 parameterized by

$$\phi_2 : \mathbb{P}^2 \to \mathbb{P}^3 : (s:t:u) \mapsto (s^3+t^3 : stu : su^2+tu^2 : u^3).$$

As in the previous example, to determine the intersection between $\mathbf{S}_1$ and $\mathbf{S}_2$ we will compute the spectrum of the polynomial matrix

$$M(s,t,u) \\ = \begin{pmatrix} -s^3-t^3 & 0 & -stu & 0 & -stu & stu & 0 & su^2+tu^2 & 0 \\ stu & -stu & 0 & u^3 & 0 & -s^3-t^3 & -stu & 0 & 0 \\ 0 & 0 & u^3 & 0 & 0 & 0 & su^2+tu^2 & 0 & -s^3-t^3 \\ u^3 & 0 & 0 & -stu & 0 & su^2+tu^2 & 0 & -stu & stu \\ 0 & u^3 & 0 & 0 & 0 & su^2+tu^2 & 0 & 0 & stu \\ u^3 & 0 & 0 & 0 & su^2+tu^2 & 0 & 0 & 0 & stu \end{pmatrix}.$$

By dehomogenizing with respect to the variable u, we consider

$$M(s,t) = \begin{pmatrix} -s^3-t^3 & 0 & -st & 0 & -st & st & 0 & s+t & 0 \\ st & -st & 0 & 1 & 0 & -s^3-t^3 & -st & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & s+t & 0 & -s^3-t^3 \\ 1 & 0 & 0 & -st & 0 & s+t & 0 & -st & st \\ 0 & 1 & 0 & 0 & 0 & s+t & 0 & 0 & st \\ 1 & 0 & 0 & 0 & s+t & 0 & 0 & 0 & st \end{pmatrix}.$$

Writing $M(s,t)$ as $M(s,t) = M_3t^3 + M_2t^2 + M_1t + M_0$, we obtain the generalized companion matrices of $M(s,t)$:

$$A(s) = \begin{pmatrix} 0 & I_6 & 0 \\ 0 & 0 & I_6 \\ M_0^t & M_1^t & M_2^t \end{pmatrix}, \quad B(s) = \begin{pmatrix} I_6 & 0 & 0 \\ 0 & I_6 & 0 \\ 0 & 0 & -M_3^t \end{pmatrix}.$$

Applying the algorithm that extracts a square matrix for the pencil $A^t(s) - tB^t(s)$, we obtain its regular part $M_1(s,t) = A_1(s) - tB_1(s)$ where

$$A_1(s) = \begin{pmatrix} s & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & s & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ s & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, B_1(s) = \begin{pmatrix} -s^2 & 1 & 0 & 0 & 0 & -s & 0 & 0 & s & 0 & 0 & 0 \\ s^3 & s & s^4 & 1 & 0 & 0 & s^3 & 0 & s^2 & 0 & 0 & 0 \\ 0 & 0 & s & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ s & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Its yields a plane curve of degree 6 whose implicit equation is $\det(M_1(s,t)) = t^2 + 2st + s^2t^2 + 2s^3t^3 - st^5 + s^2 - ts^5$. This plane curve parameterizes $\mathbf{S}_1 \cap \mathbf{S}_2$ through the regular map ϕ_2.

4.5 Matrix-Based Implicit Representations of Parametric Curves in Space

Let f_0, f_1, f_2, f_3 be homogeneous polynomials in $\mathbb{R}[s,t]$ of the same degree $d \geq 1$ such that their greatest common divisor is a non-zero constant in $\mathbb{R}$. Consider the regular map of a parametric space curve

$$\begin{aligned} \mathbb{P}^1_{\mathbb{R}} &\xrightarrow{\phi} \mathbb{P}^3_{\mathbb{R}} \\ (s:t) &\mapsto (f_0 : f_1 : f_2 : f_3)(s,t). \end{aligned}$$

4.5.1 Construction of the Representation Matrix

Consider the set of syzygies of $\mathbf{f} := (f_0, f_1, f_2, f_3)$, that is to say the set

$$\mathrm{Syz}(\mathbf{f}) = \left\{ (g_0(s,t), \ldots, g_3(s,t)) \ : \ \sum_{i=0}^{3} g_i(s,t) f_i(s,t) = 0 \right\} \subset \bigoplus_{i=0}^{3} \mathbb{R}[s,t].$$

From a classical structure theorem of commutative algebra called the Hilbert-Burch Theorem (see for instance [12, §20.4]), Syz(**f**) is known to be a *free* and graded $\mathbb{R}[s,t]$-module of rank 3. Moreover, there exists non-negative integers μ_1, μ_2, μ_3 and 3 vectors of polynomials

$$(u_{i,0}(s,t), u_{i,1}(s,t), u_{i,2}(s,t), u_{i,3}(s,t)) \in \mathrm{Syz}(\mathbf{f}) \subset \mathbb{R}[s,t]^4, \ \ i = 1,2,3, \qquad (4.5)$$

such that

- For every $i \in \{1,2,3\}$, $j \in \{0,1,2,3\}$, $u_{i,j}(s,t)$ is a homogeneous polynomial in $\mathbb{R}[s,t]$ of degree $\mu_i \geq 0$,
- The 3 vectors in (4.5) form an $\mathbb{R}[s,t]$-basis of Syz(**f**),
- $\mu_1 + \mu_2 + \mu_3 = d$ where $d = \deg f_i$.
- For every $j \in \{0,\dots,3\}$, the determinant of the matrix obtained by deleting the column $(u_{i,j})_{i=1,2,3}$ from the matrix

$$M(s,t) := \begin{pmatrix} u_{1,0}(s,t) & u_{1,1}(s,t) & u_{1,2}(s,t) & u_{1,3}(s,t) \\ u_{2,0}(s,t) & u_{2,1}(s,t) & u_{2,2}(s,t) & u_{2,3}(s,t) \\ u_{3,0}(s,t) & u_{3,1}(s,t) & u_{3,2}(s,t) & u_{3,3}(s,t) \end{pmatrix} \tag{4.6}$$

  is equal to $(-1)^j c\, f_j(s,t) \in \mathbb{R}[s,t]$ where $c \in \mathbb{R} \setminus \{0\}$.

A collection of vectors as in (4.5) that satisfy the above properties is called a μ-basis of the parameterization ϕ. It is important to notice that a μ-basis is far from unique, but the collection of integers (μ_1, μ_2, μ_3) is unique if we order it. Therefore, in the sequel we will always assume that a μ-basis is ordered so that $0 \leq \mu_1 \leq \mu_2 \leq \mu_3$.

For every integer $i = 1,2,3$ and every integer $\nu \in \mathbb{N}$, consider the matrix $\mathrm{Sylv}_\nu(u_i)$ that satisfies to the identity

$$\left[s^\nu \; s^{\nu-1}t \; \cdots \; st^{\nu-1} \; t^\nu\right] \times \mathrm{Sylv}_\nu(u_i) = \left[s^{\nu-\mu_i}u_i \; s^{\nu-\mu_i-1}tu_i \; \cdots \; st^{\nu-\mu_i-1}u_i \; t^{\nu-\mu_i}u_i\right].$$

It is a $(\nu+1) \times (\nu - \mu_i + 1)$-matrix which usually appears as a building block in well-known Sylvester matrices. It follows that the matrix

$$\mathrm{Sylv}_\nu(u_1,u_2,u_3) := \left(\mathrm{Sylv}_\nu(u_1) \,\middle|\, \mathrm{Sylv}_\nu(u_2) \,\middle|\, \mathrm{Sylv}_\nu(u_3) \right).$$

It has $\nu + 1$ rows and $3(\nu+1) - d$ columns. Its entries are *linear forms* in $\mathbb{R}[x,y,z,w]$; in particular, it can be evaluated at any point $(x:y:z:w) \in \mathbb{P}^3_{\mathbb{R}}$ and yielding a matrix with coefficients in $\mathbb{R}$.

In [8], we proved that for all $\nu \geq \mu_3 + \mu_2 - 1$ the matrix $\mathtt{M}(\phi)_\nu := \mathrm{Sylv}_\nu(u_1,u_2,u_3)$ is a *matrix-based representation* of the curve $\mathcal{C}$, i.e.,

(i) $\mathtt{M}(\phi)_\nu$ is generically full rank, that is to say generically of rank $\nu + 1$,
(ii) The rank of $\mathtt{M}(\phi)_\nu$ drops exactly on the curve $\mathcal{C}$.

Of course, in practice the most useful matrix is the smallest one, that is to say $\mathtt{M}(\phi)_{\mu_3+\mu_2-1}$.

Example 4.5 Let $\mathcal{C}$ be the rational space curve parameterized by

$$\mathbb{P}^1_{\mathbb{R}} \xrightarrow{\phi} \mathbb{P}^3_{\mathbb{R}}$$

$$(s:t) \mapsto (s^4 : s^3t : s^2t^2 : t^4).$$

A μ-basis of $\mathcal{C}$ is given by

$$\begin{aligned} p &= -tx + sy, \\ q &= -ty + sz, \\ r &= -t^2z + s^2w. \end{aligned}$$

We have $\mu_1 = \mu_2 = 1$, $\mu_3 = 2$ and hence $\mu_3 + \mu_2 - 1 = 2$. Therefore, we obtain the following representation matrix of ϕ:

$$\mathtt{M}(\phi)_2 = \begin{pmatrix} y & 0 & z & 0 & w \\ -x & y & -y & z & 0 \\ 0 & -x & 0 & -y & -z \end{pmatrix}.$$

4.5.2 Points on Curves and Inversion Problems

The same ideas introduced in Sect. 4.2.2, we can solve the points on curves and inversion problems. For instance, suppose given a parameterization ϕ of a rational curve $\mathcal{C}$ and a point P in $\mathbb{P}^3$. Denote by $\mathtt{M}(\phi)_\nu$ a matrix representation of ϕ for some integer $\nu \geq \mu_3 + \mu_2 - 1$. Since its entries are linear forms in the variables x, y, z, w, one can evaluate $\mathtt{M}(\phi)_\nu$ at P and get a matrix with coefficients in the ground field $\mathbb{R}$. Then, we have that

$$\operatorname{rank}(\mathtt{M}(\phi)_\nu(P)) < \nu + 1 \text{ if and only if } P \in \mathcal{C}.$$

This property answers the point-on curve problem.

If $\operatorname{rank}\mathtt{M}(\phi)_\nu(P) = \operatorname{rank}\mathtt{M}(\phi)_\nu - 1 = \nu$ then P has a unique pre-image $(s_0 : t_0)$ by ϕ and moreover, this pre-image can be recovered from the computation of a generator, say $W_P = (w_0, \dots, w_\nu) \in \mathbb{R}^{\nu+1}$, of the kernel of the transpose of $\mathtt{M}(\phi)_\nu(P)$. Indeed, if $b_0(s,t), \dots, b_\nu(s,t)$ is the basis which has been chosen to build $\mathtt{M}(\phi)_\nu$, then there exists $\lambda \in \mathbb{R} \setminus \{0\}$ such that

$$W_P = \lambda\,(b_0(s_0,t_0), \dots, b_\nu(s_0,t_0))\,.$$

For instance, suppose that $b_i(s,t) = s^i t^{\nu-i}$, $i = 0, \dots, \nu$ (the usual monomial basis), then $(s_0 : t_0) = (w_1 : w_0)$ if $w_0 \neq 0$, otherwise $(s_0 : t_0) = (1 : 0)$.

We point out that the points $P \in \mathcal{C}$ such that $\operatorname{rank}\mathtt{M}(\phi)_\nu(P) = \operatorname{rank}\mathtt{M}(\phi)_\nu - 1 = \nu$ are precisely the regular points on $\mathcal{C}$, that is to say that all the points that do not verify this property are singular points on $\mathcal{C}$. We will come back again to this property and to the treatment of the singular points on $\mathcal{C}$ in the next section.

Example 4.6 Suppose that the parameterization ϕ is given by

$$\begin{aligned}
f_0(s,t) &= 3s^4t^2 - 9s^3t^3 - 3s^2t^4 + 12st^5 + 6t^6,\\
f_1(s,t) &= -3s^6 + 18s^5t - 27s^4t^2 - 12s^3t^3 + 33s^2t^4 + 6st^5 - 6t^6,\\
f_2(s,t) &= s^6 - 6s^5t + 13s^4t^2 - 16s^3t^3 + 9s^2t^4 + 14st^5 - 6t^6,\\
f_3(s,t) &= -2s^4t^2 + 8s^3t^3 - 14s^2t^4 + 20st^5 - 6t^6.
\end{aligned}$$

A μ-basis for $\mathcal{C}$ is

$$\begin{aligned}
p &= (s^2 - 3st + t^2)x + t^2y,\\
q &= (s^2 - st + 3t^2)y + (3s^2 - 3st - 3t^2)z,\\
r &= 2t^2z + (s^2 - 2st - 2t^2)w.
\end{aligned}$$

From $\deg(p) = \deg(q) = \deg(r) = 2$, we have $\mu_3 + \mu_2 - 1 = 3$ and hence a matrix representation of $\mathcal{C}$ is given by

$$\mathtt{M}(\phi)_3 = \begin{pmatrix} x+y & 0 & 3y-3z & 0 & 2z-2w & 0\\ -3x & x+y & -y-3z & 3y-3z & -2w & 2z-2w\\ x & -3x & y+3z & -y-3z & w & -2w\\ 0 & x & 0 & y+3z & 0 & w \end{pmatrix}.$$

Let $P = (1:1:1:1) \in \mathbb{P}^3$. Evaluating $\mathtt{M}(\phi)_3$ at P we find that

$$\mathtt{M}(\phi)_3 = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0\\ -3 & 2 & -4 & 0 & -2 & 0\\ 1 & -3 & 4 & -4 & 1 & -2\\ 0 & 1 & 0 & 4 & 0 & 1 \end{pmatrix}$$

is of rank 4 so that P does not lie on $\mathcal{C}$.

If one evaluates the matrix $\mathtt{M}(\phi)_3$ at the point $P = (9:9:9:6) \in \mathbb{P}^3$ we obtain the matrix

$$\mathtt{M}(\phi)_3(P) = \begin{pmatrix} 18 & 0 & 0 & 0 & 6 & 0\\ -27 & 18 & -36 & 0 & -12 & 6\\ 9 & -27 & 36 & -36 & 6 & -12\\ 0 & 9 & 0 & 36 & 0 & 6 \end{pmatrix}.$$

which has rank 3. Therefore, P is a smooth point on the curve $\mathcal{C}$. Moreover, the computation of the kernel of the transpose of $\mathtt{M}(\phi)_3(P)$ returns the vector $(1,1,1,1)$. Thus, we deduce that $P = \phi(1:1)$.

4.5.3 Rank of a Representation Matrix at a Singular Point

Let P be a point on $\mathcal{C}$. There exists at least one point $(s_1 : t_1) \in \mathbb{P}^1$ such that $P = \phi(s_1 : t_1)$. Now, let $\mathcal{H}$ be a plane in $\mathbb{P}^3$ passing through P, not containing $\mathcal{C}$ and denote by $H(x, y, z, w)$ an equation (a linear form in $\mathbb{R}[x, y, z, w]$) of $\mathcal{H}$. We have the following degree d homogeneous polynomial in $\mathbb{R}[s, t]$

$$H(f_0(s,t), f_1(s,t), f_2(s,t), f_3(s,t)) = \prod_{i=1}^{d}(t_i s - s_i t) \tag{4.7}$$

where the points $(s_i : t_i) \in \mathbb{P}^1$, $i = 1, \ldots, d$, are not necessarily distinct. We define the intersection multiplicity of $\mathcal{C}$ with $\mathcal{H}$ at the point P, denoted $i_P(\mathcal{C}, \mathcal{H})$, as the number of points $(s_i : t_i)_{i=1,\ldots,d}$ such that $\phi(s_i : t_i) = P$.

The *multiplicity* $m_P(\mathcal{C})$ of the point P on $\mathcal{C}$ is defined as the minimum of the intersection multiplicities $i_P(\mathcal{C}, \mathcal{H})$ where $\mathcal{H}$ runs over all the planes not containing $\mathcal{C}$ and passing through the point $P \in \mathcal{C}$. This minimum is reached for a sufficiently generic $\mathcal{H}$.

Suppose given a representation matrix $\mathtt{M}(\phi)_\nu$ of the curve $\mathcal{C}$ which is built from the μ-basis p, q, r of degree $\mu_1 \le \mu_2 \le \mu_3$. Its entries are linear forms in $\mathbb{R}[x, y, z, w]$ so that it makes sense to evaluate $\mathtt{M}(\phi)_\nu$ at a point P in $\mathbb{P}^3$ to get a matrix $\mathtt{M}(\phi)_\nu(P)$ with entries in $\mathbb{R}$. In [8], we prove the following property: Given a point P in $\mathbb{P}^3$, for every integer $\nu \ge \mu_2 + \mu_3 - 1$ we have

$$\operatorname{rank} \mathtt{M}(\phi)_\nu(P) = \nu + 1 - m_P(\mathcal{C}),$$

or equivalently $\operatorname{corank} \mathtt{M}(\phi)_\nu(P) = m_P(\mathcal{C})$.

This result provides a stratification of the points in $\mathbb{P}^3$ with respect to the curve $\mathcal{C}$. Indeed, we have that

- If P is such that $\operatorname{rank} \mathtt{M}(\phi)_\nu(P) = \nu + 1$ then $P \notin \mathcal{C}$,
- If P is such that $\operatorname{rank} \mathtt{M}(\phi)_\nu(P) = \nu$ then P is a regular point (i.e. of multiplicity 1) on $\mathcal{C}$,
- If P is such that $\operatorname{rank} \mathtt{M}(\phi)_\nu(P) = \nu - 1$ then P is singular point of multiplicity 2 on $\mathcal{C}$,
- and so on.

Moreover, if P is a singular point on $\mathcal{C}$ then necessarily

$$2 \le m_P(\mathcal{C}) \le \mu_2 \text{ or } m_P(\mathcal{C}) = \mu_3. \tag{4.8}$$

One can read more details in [8] for computational singularities aspects of $\mathcal{C}$.

4.5.4 Curve/Curve Intersection

Suppose given two rational curves, say $\mathcal{C}_1$ parameterized by

$$\mathbb{P}^1 \xrightarrow{\phi_1} \mathbb{P}^3 : (s:t) \mapsto (f_0 : \cdots : f_3)(s,t) \tag{4.9}$$

and $\mathcal{C}_2$ parameterized by the regular map

$$\mathbb{P}^1 \xrightarrow{\phi_2} \mathbb{P}^3 : (s:t) \mapsto (g_0 : \cdots : g_3)(s,t). \tag{4.10}$$

Let $\mathtt{M}(\phi_1)_\nu$ be a representation matrix of $\mathcal{C}_1$ for a suitable integer ν. The substitution in $\mathtt{M}(\phi_1)_\nu$ of the variables x, y, z, w by the homogeneous parameterization of $\mathcal{C}_2$ yields the matrix

$$\mathtt{M}(\phi_1)_\nu(s,t) := \mathtt{M}(\phi_1)_\nu(g_0(s,t), \ldots, g_3(s,t)).$$

As a consequence of the properties of a representation matrix, we have the following property: Let $(s_0 : t_0) \in \mathbb{P}^1$. Then $\operatorname{rank} \mathtt{M}(\phi_1)_\nu(s_0,t_0) < \nu + 1$ if and only if the point $\phi_2(s_0,t_0)$ belongs to the intersection locus $\mathcal{C}_1 \cap \mathcal{C}_2$.

The set $\mathcal{C}_1 \cap \mathcal{C}_2$ is in correspondence with the points of $\mathbb{P}^1$ where the rank of $\mathtt{M}(\phi_1)_\nu(s,t)$ drops. By setting $t = 1$, the determination of the values of s such that the rank of $\mathtt{M}(\phi_1)_\nu(s,1)$ drops can be treated at the level of matrices (that is to say without any symbolic computation and in particular without any determinant computations) by using linearization techniques and generalized eigenvalues computations. We obtain the algorithm similar to Algorithm 1.

Algorithm 3: Intersection of two parametric curves

```
Input: Two parametric curves $\mathcal{C}_1$ and $\mathcal{C}_2$ given by (4.9) and (4.10).
Output: The intersection points of $\mathcal{C}_1$ and $\mathcal{C}_2$.
1. Compute the matrix representation $\mathtt{M}(\phi)_\nu(\phi_1)$ of $\mathcal{C}_1$ for a suitable $\nu$.
2. Compute the generalized companion matrices $A$ and $B$ of $\mathtt{M}(\phi)_\nu(\phi_1)$.
3. Compute the companion regular matrices $A'$ and $B'$.
4. Compute the generalized eigenvalues of $(A', B')$.
5. For each eigenvalue $t_0$, $\phi_2(t_0 : 1)$ is an intersection point.
```

Remark that this algorithm returns all the points in $\mathcal{C}_1 \cap \mathcal{C}_2$ except possibly the point $\phi(1:0)$. This point can be treated independently.

Conclusion

This paper presents an implicit representation concept of a parametric curve or a parametric surface. This representation is a matrix whose entries are linear forms in the coordinates of $\mathbb{R}^3$. This matrix representation characterizes a

(continued)

curve or a surface by a rank drop property. It is easily to calculate and in addition a useful tool for solving intersection problems. Moreover, its main interest is particularly to transform intersection problems into numerical linear algebra problems which can be solved using powerful and robust algorithms, such as the singular value decomposition and the computation of generalized eigenvalues or eigenvectors. Thus, in the context of ray tracing on a surface set, this approach could improve the robustness of the existing methods in particular situations.

All algorithms that we proposed above have been implemented in the software `Maple` and the corresponding files are available at http://cgi.di.uoa.gr/~thanglb/.

Acknowledgements Part of this work has been completed while the author worked at the Department of Informatics and Telecommunications, University of Athens, Greece with partial support from Marie-Curie Initial Training Network "SAGA" (ShApes, Geometry, Algebra), FP7-PEOPLE contract PITN-GA-2008-214584.

References

1. Th. Beelen, P. Van Dooren. An improved algorithm for the computation of Kronecker's canonical form of a singular pencil. Linear Algebra Appl. **105**, 9–65 (1988)
2. N. Botbol, M. Dohm, A. Dickenstein, Matrix representations for toric parametrizations. Comput. Aided Geom. Des. **7**, 757–771 (2009)
3. L. Busé, M. Chardin, Implicitizing rational hypersurfaces using approximation complexes. J. Symb. Comput. **40**, 1150–1168 (2005)
4. L. Busé, M. Chardin, and with appendix of J. Oesterlé Aron Simis. Elimination and nonlinear equations of rees algebras. J. Algebra **324**(3), 1314–1333 (2010)
5. L. Busé, M. Chardin, J.-P. Jouanolou. Torsion of the symmetric algebra and implicitization. Proc. Am. Math. Soc. **137**(06), 1855–1865 (2009)
6. L. Busé, D. Cox, C. D'Andrea, Implicitization of surfaces in $\mathbb{P}^3$ in the presence of base points. J. Algebra Appl. **2**(2), 189–214 (2003)
7. L. Busé, M. Dohm, Implicitization of bihomogeneous parametrizations of algebraic surfaces via linear syzygies, in *International Symposium on Symbolic and Algebraic Computation, ISSAC 2007*, Waterloo ACM, New York, pp. 69–76 (2007)
8. L. Busé, T. Luu Ba, Matrix-based implicit representations of algebraic curves and applications. Comput. Aided Geom. Des. **27**(9), 681–699 (2010)
9. L. Busé, T. Luu Ba. The surface/surface intersection problem by means of matrix based representations. Comput. Aided Geom. Des. **29**, 579–598 (2012)
10. L. Busé, J.-P. Jouanolou, On the closed image of a rational map and the implicitization problem. J. Algebra **265**(1), 312–357 (2003)
11. D. Cox, R. Goldman, M. Zhang, On the validity of implicitization by moving quadrics of rational surfaces with no base points. J. Symb. Comput. **29**(3), 419–440 (2000)
12. D. Eisenbud, *Commutative Algebra with a View Toward Algebraic Geometry*. Graduate Text in Mathematics, vol. 150 Springer, New York, 1995

13. F.R. Gantmacher, *Théorie des matrices. Tome 2: Questions spéciales et applications*. Traduit du Russe par Ch. Sarthou. Collection Universitaire de Mathématiques, No. 19 Dunod, Paris, 1966
14. I. Gohberg, P. Lancaster, L. Rodman, *Matrix Polynomial*, vol. 58 SIAM, Philadelphia, 2009
15. G.H. Golub, C.F. Van Loan, *Matrix Computaions*. Johns Hopkins Studies in the Mathematical Sciences, 3rd edn. Johns Hopkins University Press, Baltimore, 1996
16. A. Khetan, C. D´Andrea, Implicitization of rational surfaces using toric varieties. J. Algebra **303**(2), 543–565 (2006)
17. V.N. Kublanovskaya, Methods and algorithm of solving spectral problems for polynomial matrices and rational matrix. J. Math. Sci. **96**(3), 3085–3287 (1999)
18. V.N. Kublanovskaya, V.B. Khazanov, Spectral problems for pencils of polynomial matrices. Methods and algorithms. J. Math. Sci. **79**(3), 1048–1076 (1996)
19. T. Luu Ba, Matrix-based implicit representations of algebraic curves and surfaces and applications, PhD thesis, University of Nice Sophia Anitpolis – INRIA Sophia Antipolis, July 2011
20. T. Luu Ba, L. Busé, B. Mourrain, Curve/surface intersection problem by means of matrix representation, in *SNC'09: Proceedings of the International Conference on Symbolic Numeric Computation*, Kyoto (ACM, NewYork, 2009), pp. 71–78
21. D. Manocha, J. Canny, A new approach for surface intersection, in *Proceedings of the First ACM Symposium on Solid Modeling Foundations and CAD/CAM Applications*, Austin, pp. 209–219 (1991)
22. B. Mourrain, Bezoutian and quotient ring structure. J. Symb. Comput. **39**(3), 397–415 (2005)
23. T.W. Sederberg, F. Chen, Implicitization using moving curves and surfaces. Proc. SIGGRAPH **29**, 301–308 (1995)
24. T.W. Sederberg, S.R. Parry, Comparison of three curve intersection algorithms. Comput. Aided Des. **18**(1), 58–63 (1986)

Part II
Geometric Computing: Algebraic Tools

Chapter 5
Singular Zeros of Polynomial Systems

Angelos Mantzaflaris and Bernard Mourrain

5.1 Introduction

A main challenge in algebraic and geometric computing is singular point identification and treatment. Such problems naturally occur when computing the topology of implicit curves or surfaces [1], the intersection of parametric surfaces in geometric modeling. When algebraic representations are used, this reduces to solving polynomial systems. Several approaches are available: algebraic techniques such as Gröbner bases or border bases, resultants, subdivision algorithms [16, 19], homotopies, and so on. At the end of the day, a numerical approximation or a box of isolation is usually computed to identify every real root of the polynomial system. But we often need to improve the numerical approximation of the roots. Numerical methods such as Newton's iteration can be used to improve the quality of the approximation, provided that we have a simple root. In the presence of a multiple root, the difficulties are significantly increasing. The numerical approximation can be of very bad quality, and the methods used to compute this approximation are converging slowly (or not converging). The situation in practical problems, as encountered in CAGD for instance, is even worse, since the coefficients of the input equations are known with some incertitude. Computing multiple roots and root multiplicities of approximate polynomial systems is an ill-posed problem, since changing slightly the coefficients may transform a multiple root into a cluster of simple roots (or even make it disappear).

For instance Newton's method converges only linearly to such a point, if it converges at all [6]. Also, certification tests for the existence of roots on a domain

A. Mantzaflaris (✉) • B. Mourrain
GALAAD, INRIA Méditerranée, BP 93, 06902 Sophia-Antipolis, France
e-mail: Angelos.Mantzaflaris@oeaw.ac.at; Angelos.Mantzaflaris@inria.fr;
Bernard.Mourrain@inria.fr

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_5

do not directly treat these cases. On the other hand, computing the local multiplicity structure around a singularity breaks down to stable linear algebra methods, which can be run approximately. One can use this local structure to deflate the root, and thus restore super-linear convergence of Newton iteration, or use standard verification techniques to certify a singular root of the original system. In case of inexact coefficients, known up to a certain tolerance, an exact singular root no longer exists. Nevertheless, a well chosen symbolic perturbation, combined with deflation, allows the certification of a nearby system, within a controlled neighborhood of the original one, which attains a single singular point.

The numerical treatment of singular zeroes is a difficult task, mainly because of the ill-posedness of the problem. The following strategy can however be adopted. Find a perturbation of the input system such that the root is a deformation of an exact multiple root. Certainly, there is not a single multiple system, if the input data is approximate. But using the knowledge of the dual structure and interval arithmetic, our method aims at providing a controlled deformation that is compatible with the input.

In this way, we identify the multiplicity structure and we are able to setup deflation techniques which restore the quadratic convergence of the Newton system. The certification of the multiple root is also possible on the symbolically perturbed system by applying a fixed point theorem, based e.g. on interval arithmetic [22] or α-theorems ([7] and references therein).

This approach has already been explored in the past. The first algebraic work on the analysis of singular points may be due to F.S. Macaulay [14], who introduced the terminology of “inverse system”. His so-called *dialytic method* has been exploited in [4, 12, 13] to construct the inverse system of a multiple point.

Another construction of inverse systems is described e.g. in [17], reducing the size of the intermediate linear systems (and exploited in [23]).

In [18], another approach to construct the dual basis at the singular point which is based on an integration strategy, has been proposed.

Regarding deflation techniques, in [20], by applying a triangulation preprocessing step on the Jacobian matrix at the approximate root, minors of the Jacobian matrix are added to the system to reduce the multiplicity.

In [11], a representation of the ideal in a triangular form in a good position and derivations with respect to the leading variables are used to iteratively reduce the multiplicity. This process is applied for p-adic lifting with exact computation.

In [12, 13], instead of triangulating the Jacobian matrix, the number of variables is doubled and new equations are introduced, which are linear in the new variables. They describe the kernel of the Jacobian matrix at the multiple root. The process is iterative, yet for some practical applications, the root may already be deflated with a few iterations.

In [4], the deflation method is applied iteratively until the root becomes regular, doubling each time the number of variables.

In [21], a minimization approach is used to reduce the value of the equations and their derivatives at the approximate root, assuming a basis of the inverse system is known.

In [24], the inverse system is constructed via Macaulay's method; tables of multiplications are deduced and their eigenvalues are used to improve the approximated root. They show that the convergence is quadratic at the multiple root.

Verification of multiple roots of (approximate) polynomial equations is a difficult task. The approach proposed in [22] consists of introducing perturbation parameters and to certifying the multiple root of nearby system by using a fixed point theorem, based on interval arithmetic. It applies only to cases where the Jacobian has corank equal to 1.

The goal of this paper is to review different techniques that can be used to handle efficiently the following tasks:

(a) Compute a basis for the dual space and of the local quotient ring at a given (approximate) singular point.
(b) Deflate the system by augmenting it with new equations derived from the dual basis, introducing adequate perturbation terms.
(c) Certify the singular point and its multiplicity structure for the perturbed system checking the contraction property of Newton iteration (e.g. via interval arithmetic).

These tools can be applied to improve the quality of approximation of a multiple isolated solution of a system of (polynomial) equations, but they can also be used to solve geometrical problems, such as for instance computing the number of real branches at a singular point of an algebraic curve. For more details on these applications, we refer to [15] and references therein.

5.2 Preliminary Considerations

In this section we present some definitions together with the main tools that we shall need in the sequel.

We denote by $R = \mathbb{K}[\boldsymbol{x}]$, $\boldsymbol{x} = (x_1, \dots, x_n)$, a polynomial ring over the field $\mathbb{K}$ of characteristic zero. The *dual ring* R^* is the space of linear functionals $\Lambda : R \to \mathbb{K}$. It is commonly identified to the space of formal series $\mathbb{K}[[\boldsymbol{\partial}]]$ where $\boldsymbol{\partial} = (\partial_1, \dots, \partial_n)$ are formal variables. Thus we view dual elements as formal series in differential operators at a point $\boldsymbol{\zeta} \in \mathbb{K}^n$. To specify that we use the point $\boldsymbol{\zeta}$, we also denote these differentials $\boldsymbol{\partial}_{\zeta}$. When applying $\Lambda(\boldsymbol{\partial}_{\zeta}) \in \mathbb{K}[[\boldsymbol{\partial}_{\zeta}]]$ to a polynomial $g(\boldsymbol{x}) \in R$ we will denote by $\Lambda^{\zeta}[g] = \Lambda^{\zeta} g = \Lambda(\boldsymbol{\partial}_{\zeta})[g(\boldsymbol{x})]$ the operation

$$\Lambda^{\zeta}[g] = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^n} \frac{\lambda_{\boldsymbol{\alpha}}}{\alpha_1! \cdots \alpha_n!} \cdot \frac{d^{|\boldsymbol{\alpha}|} g}{dx_1^{\alpha_1} \cdots dx_n^{\alpha_n}}(\boldsymbol{\zeta}), \tag{5.1}$$

for $\Lambda(\boldsymbol{\partial}_{\zeta}) = \sum \lambda_{\boldsymbol{\alpha}} \frac{\mathbf{1}}{\boldsymbol{\alpha}!} \boldsymbol{\partial}_{\zeta}^{\boldsymbol{\alpha}} \in \mathbb{K}[[\boldsymbol{\partial}_{\zeta}]]$. Extending this definition to an ordered set $\boldsymbol{\mathcal{D}} = (\Lambda_1, \dots, \Lambda_{\mu}) \in \mathbb{K}[[\boldsymbol{\partial}]]^{\mu}$, we shall denote $\boldsymbol{\mathcal{D}}^{\zeta}[g] = (\Lambda_1^{\zeta} g, \dots, \Lambda_{\mu}^{\zeta} g)$. In some cases, it is convenient to use normalized differentials instead of $\boldsymbol{\partial}$: for any $\boldsymbol{\alpha} \in \mathbb{N}^n$, we denote $\boldsymbol{d}_{\zeta}^{\boldsymbol{\alpha}} = \frac{1}{\boldsymbol{\alpha}!}\boldsymbol{\partial}_{\zeta}^{\boldsymbol{\alpha}}$. In particular, with the use of this notation we recover the nice property that, if $\boldsymbol{\zeta} = \mathbf{0}$, we have $\boldsymbol{d}_{\mathbf{0}}^{\boldsymbol{\alpha}}\boldsymbol{x}^{\boldsymbol{\beta}} = 1$ if $\boldsymbol{\alpha} = \boldsymbol{\beta}$ and 0 otherwise.

More generally, $(\boldsymbol{d}_{\zeta}^{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathbb{N}^n}$ is the dual basis of $((\boldsymbol{x} - \boldsymbol{\zeta})^{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathbb{N}^n}$, i.e., a non-zero root implies a linear transformation of the variables, so that the root is translated to $(0, 0)$.

Example 5.1 Consider the integral of a polynomial function $g \in R$ over the unit hypercube. Since this is a linear map, it may be expressed in terms of differentials, i.e.:

$$g \longmapsto \int_{[0,1]^n} g(\boldsymbol{x})\,dx_1 \cdots dx_n = \sum_{\boldsymbol{\alpha}\in\sup(g)} c_{\boldsymbol{\alpha}}\,\boldsymbol{d}^{\boldsymbol{\alpha}}[g]\ ,$$

where $\boldsymbol{d}^{\boldsymbol{\alpha}}[g] = \frac{1}{\boldsymbol{\alpha}!}\frac{\partial^{|\boldsymbol{\alpha}|}g}{\boldsymbol{\partial x}^{\boldsymbol{\alpha}}}(\mathbf{0})$ and $\sup(g)$ stands for the support of g. Indeed, it can be verified using simple calculations that the (unique) coefficients are given by $c_{\boldsymbol{\alpha}} = \prod_{i=1}^{n}\frac{1}{\alpha_i + 1}$.

For $\Lambda \in R^*$ and $p \in R$, let us define the operation $p \cdot \Lambda : q \mapsto \Lambda(p\,q)$. We check that

$$(x_i - \zeta_i) \cdot \boldsymbol{\partial}_{\zeta}^{\boldsymbol{\alpha}} = \frac{d}{d\,\partial_{i,\zeta}}(\boldsymbol{\partial}_{\zeta}^{\boldsymbol{\alpha}}), \tag{5.2}$$

and R^* obtains the structure of an $R-$module. This property shall be useful in the sequel.

5.2.1 Isolated Points and Differentials

Let $\mathcal{I} = \langle f_1, \ldots, f_s\rangle$ be an ideal of R and $\boldsymbol{\zeta} \in \mathbb{K}^n$ a root of the polynomial system $\boldsymbol{f} = (f_1, \ldots, f_s)$. We call $\boldsymbol{\zeta}$ an isolated zero of $V(\mathcal{I})$ if, in a primary decomposition of $\mathcal{I}$, the radical of one of the primary components is the maximal ideal $m_{\zeta} = \langle x_1 - \zeta_1, \ldots, x_n - \zeta_n\rangle$ defining $\boldsymbol{\zeta}$ and no other primary component is contained in m_{ζ}.

Suppose that $\boldsymbol{\zeta}$ is an isolated root of $\boldsymbol{f}$, then a minimal primary decomposition of

$$\mathcal{I} = \bigcap_{\mathcal{Q}\ \text{prim.}\supset\mathcal{I}} \mathcal{Q}$$

contains a primary component $\mathcal{Q}_\zeta$ such that $\sqrt{\mathcal{Q}_\zeta} = m_\zeta$ and $\sqrt{\mathcal{Q}'} \not\subset m_\zeta$ for the other primary components $\mathcal{Q}'$ associated to $\mathcal{I}$ [2].

As $\sqrt{\mathcal{Q}_\zeta} = m_\zeta$, it follows that $R/\mathcal{Q}_\zeta$ is a finite dimensional vector space. The multiplicity μ_ζ of $\boldsymbol{\zeta}$ is defined as the dimension of $R/\mathcal{Q}_\zeta$. A point of multiplicity one is called *regular point*, or *simple root*, otherwise we say that $\boldsymbol{\zeta}$ is a singular isolated point, or multiple root of $\boldsymbol{f}$. In the latter case we have $J_{\boldsymbol{f}}(\boldsymbol{\zeta}) = 0$.

Example 5.2 Consider the ideal $\mathcal{I} = \langle x_1 - x_2 + x_1^2, x_1 - x_2 + x_2^2 \rangle$, and the root $\boldsymbol{\zeta} = (0, 0)$. Then a minimal primary decomposition of $\mathcal{I}$ is

$$\mathcal{I} = \langle x_2^3, x_1 - x_2 + x_2^2 \rangle \cap \langle -2 + x_2, 2 + x_1 \rangle \ .$$

Among the two factors we find the maximal ideal of $\boldsymbol{\zeta}$ given by the radical ideal $\sqrt{\langle x_2^3, x_1 - x_2 + x_2^2 \rangle} = \langle x_1, x_2 \rangle$.

We can now define the dual space of an ideal.

Definition 5.1 *The* dual space *of $\mathcal{I}$ is the subspace of elements of* $\mathbb{K}[[\boldsymbol{\partial}_\zeta]]$ *(formal series of the variables $\boldsymbol{\partial}_\zeta$), $\boldsymbol{\zeta} \in V(\mathcal{I})$, that vanish on all the elements of $\mathcal{I}$. It is also called the orthogonal of $\mathcal{I}$ and is denoted as $\mathcal{I}^\perp$.*

The dual space is known to be isomorphic to the quotient $R/\mathcal{I}$. Consider now the *orthogonal* of $\mathcal{Q}_\zeta$, i.e. the subspace $\mathscr{D}_\zeta$ of elements of R^* that vanish on members of $\mathcal{Q}_\zeta$, namely

$$\mathcal{Q}_\zeta^\perp = \mathscr{D}_\zeta = \{\Lambda \in R^* \ : \ \Lambda^\zeta[p] = 0, \ \forall p \in \mathcal{Q}_\zeta\}.$$

The following is an essential property that allows extraction of the local structure $\mathscr{D}_\zeta$ directly from the "global" ideal $\mathcal{I} = \langle \boldsymbol{f} \rangle$, notably by matrix methods that will be outlined in Sect. 5.3.

Proposition 5.1 ([18, Th. 8]) *For any isolated point $\boldsymbol{\zeta} \in \mathbb{K}$ of $\boldsymbol{f}$, we have $\mathcal{I}^\perp \cap \mathbb{K}[\boldsymbol{\partial}_\zeta] = \mathscr{D}_\zeta$.*

In other words, we can identify $\mathscr{D}_\zeta = \mathcal{Q}_\zeta^\perp$ with the space of polynomial differential operators that vanish at $\boldsymbol{\zeta}$ on every element of $\mathcal{I}$. Also note that $\mathscr{D}_\zeta^\perp = \mathcal{Q}_\zeta$.

The space $\mathscr{D}_\zeta$ has dimension μ_ζ, the multiplicity at $\boldsymbol{\zeta}$. As the variables $(x_i - \zeta_i)$ act on R^* as derivations (see (5.2)), $\mathscr{D}_\zeta$ is a space of differential polynomials in $\boldsymbol{\partial}_\zeta$, which is stable under derivation. This property will be used explicitly in constructing $\mathscr{D}_\zeta$ (Sect. 5.3).

Definition 5.2 *The* nilindex *of $\mathcal{Q}_\zeta$ is the maximal integer $N \in \mathbb{N}$ such that $m_\zeta^N \not\subset \mathcal{Q}_\zeta$.*

It is directly seen that the maximal order of elements in $\mathscr{D}_\zeta$ is equal to N, also known as the *depth* of the space.

5.2.2 Quotient Ring and Dual Structure

In this section we explore the relation between the dual ring and the quotient $R/\mathcal{Q}_{\boldsymbol{\zeta}}$, where $\mathcal{Q}_{\boldsymbol{\zeta}}$ is the primary component of the isolated point $\boldsymbol{\zeta}$. We show how to extract a basis of this quotient ring from the support of the elements of $\mathscr{D}_{\boldsymbol{\zeta}}$ and how $\mathscr{D}_{\boldsymbol{\zeta}}$ can be used to reduce any polynomial modulo $\mathcal{Q}_{\boldsymbol{\zeta}}$.

It is convenient in terms of notation to make the assumption $\boldsymbol{\zeta} = \mathbf{0}$. This poses no constraint, since it implies only a linear change of coordinates.

Let $\operatorname{supp}\mathscr{D}_{\mathbf{0}}$ be the set of exponents of monomials appearing in $\mathscr{D}_{\mathbf{0}}$, with a non-zero coefficient. These are of degree at most N, the nilindex of $\mathcal{Q}_{\mathbf{0}}$. Since

$$(\forall \Lambda \in \mathscr{D}_{\mathbf{0}},\ \Lambda^{\mathbf{0}}[p] = 0) \text{ iff } p \in \mathscr{D}_{\mathbf{0}}^{\perp} = \mathcal{Q}_{\mathbf{0}}\ ,$$

we derive that $\operatorname{supp}\mathscr{D}_{\mathbf{0}} = \{\boldsymbol{\gamma} : \boldsymbol{x}^{\boldsymbol{\gamma}} \notin \mathcal{Q}_{\mathbf{0}}\}$. In particular, we can find a basis of $R/\mathcal{Q}_{\mathbf{0}}$ between the monomials $\{\boldsymbol{x}^{\boldsymbol{\gamma}} : \boldsymbol{\gamma} \in \operatorname{supp}\mathscr{D}_{\mathbf{0}}\}$. This is a finite set of monomials, since their degree is bounded by the nilindex of $\mathcal{Q}_{\mathbf{0}}$. Now let Let $\boldsymbol{x}^{\boldsymbol{\gamma}_j}$, $j = 1, \dots, s$ be an enumeration of these monomials. It is clear that these are finitely many, since $\mathcal{Q}_{\mathbf{0}}$ is zero-dimensional. Given a monomial basis $\mathcal{B} = (\boldsymbol{x}^{\boldsymbol{\beta}_i})_{i=1,\dots,\mu}$ of $R/\mathcal{Q}_{\mathbf{0}}$ and, for all monomials $\boldsymbol{x}^{\boldsymbol{\gamma}_j} \notin \mathcal{Q}_{\mathbf{0}}$, the expression (normal form)

$$x^{\boldsymbol{\gamma}_j} = \sum_{i=1}^{\mu} \lambda_{ij} \boldsymbol{x}^{\boldsymbol{\beta}_i} \pmod{\mathcal{Q}_{\mathbf{0}}} \tag{5.3}$$

of $x^{\boldsymbol{\gamma}_j}$ in the basis $\mathcal{B}$, then the dual elements [18, Prop. 13]

$$\Lambda_i(\boldsymbol{d}) = \boldsymbol{d}^{\boldsymbol{\beta}_i} + \sum_{j=1}^{s-\mu} \lambda_{ij} \boldsymbol{d}^{\boldsymbol{\gamma}_j}, \tag{5.4}$$

for $i = 1, \dots, \mu$ form a basis of $\mathscr{D}_{\boldsymbol{\zeta}}$. We give a proof of this fact in the following lemma.

Lemma 5.1 *The set of elements $\mathcal{D} = (\Lambda_i)_{i=1,\dots,\mu}$ defined in (5.4) is a basis of $\mathscr{D}_{\boldsymbol{\zeta}}$ and the normal form of any $g(\boldsymbol{x}) \in R$ with respect to the monomial basis $\mathcal{B} = (\boldsymbol{x}^{\boldsymbol{\beta}_i})_{i=1,\dots,\mu}$ is*

$$NF(g) = \sum_{i=1}^{\mu} \Lambda_i^{\boldsymbol{\zeta}}[g]\, \boldsymbol{x}^{\boldsymbol{\beta}_i}. \tag{5.5}$$

Proof First note that the elements of $\mathcal{D}$ are linearly independent, since $\boldsymbol{d}^{\boldsymbol{\beta}_i}$ appears only in $\Lambda_i(\boldsymbol{d})$. Now, by construction,

$$\sum_{i=1}^{\mu} \Lambda_i^{\zeta}[\boldsymbol{x}^{\boldsymbol{\alpha}}]\boldsymbol{x}^{\boldsymbol{\beta}_i} = \mathrm{NF}(\boldsymbol{x}^{\boldsymbol{\alpha}}) \ ,$$

for all $\boldsymbol{x}^{\boldsymbol{\alpha}} \notin \mathcal{Q}_\zeta$, e.g. $\mathrm{NF}(\boldsymbol{x}^{\boldsymbol{\beta}_i}) = \boldsymbol{x}^{\boldsymbol{\beta}_i}$. Also, for $\boldsymbol{x}^{\boldsymbol{\alpha}} \in \mathcal{Q}_\zeta$, $\forall i$, $\Lambda_i^{\zeta}(\boldsymbol{x}^{\boldsymbol{\alpha}}) = 0$, since $\boldsymbol{\alpha} \notin \operatorname{supp}\mathcal{D}$. Thus the elements of $\mathcal{D}$ compute $\mathrm{NF}(\cdot)$ on all monomials of R, and (5.5) follows by linearity. We deduce that $\mathcal{D}$ generates the dual, as in Definition 5.1. □

It becomes clear that with the knowledge of the dual basis at $\boldsymbol{\zeta}$, we are able to compute any $g \in R$ modulo $\mathcal{Q}_\zeta$ by applying the basis elements to the monomials of g (formal derivation plus evaluation at $\boldsymbol{\zeta}$). This lemma also shows an isomorphism between the dual $\mathscr{D}_\zeta$ and the quotient ring $R/\mathcal{Q}_\zeta$, since it implies a one-to-one mapping between the primal and dual basis.

Example 5.3 Consider $f(x,y) = x^4 + 2x^2y^2 + y^4 + 3x^2y - y^3$ and $g(x,y) = 18xy^2 - 6x^3$. The common zero $\boldsymbol{\zeta} = (0,0)$ yields the local dual space

$$\mathcal{D} = (1, d_x, d_y, d_x{}^2, d_x d_y, d_y{}^2, d_x{}^3 + \tfrac{1}{3}\, d_x d_y{}^2, d_x{}^2 d_y + 3\, d_y{}^3,$$
$$d_x{}^4 + \tfrac{1}{3}\, d_x{}^2 d_y{}^2 + d_y{}^4 + \tfrac{8}{3}\, d_y^3),$$

therefore $\boldsymbol{\zeta}$ is a singular zero with multiplicity $m = 9$.

The primal counterpart is $\mathcal{B} = (1, x, y, x^2, xy, y^2, x^3, x^2y, x^4)$. The relation between $\mathcal{D}$ and $\mathcal{B}$ is revealed in the following construction:

$$\begin{array}{c|ccccccccccccc|}
 & 1 & x & y & x^2 & xy & y^2 & x^3 & x^2y & x^4 & xy^2 & y^3 & x^2y^2 & y^4 \\
\hline
1 & 1 & & & & & & & & & 0 & 0 & 0 & 0 \\
d_x & & 1 & & & & & & & & 0 & 0 & 0 & 0 \\
d_y & & & 1 & & & & & & & 0 & 0 & 0 & 0 \\
d_x^2 & & & & 1 & & & & & & 0 & 0 & 0 & 0 \\
d_xd_y & & & & & 1 & & & & & 0 & 0 & 0 & 0 \\
d_y^2 & & & & & & 1 & & & & 0 & 0 & 0 & 0 \\
d_x^3 & & & & & & & 1 & & & 1/3 & 0 & 0 & 0 \\
d_x^2d_y & & & & & & & & 1 & & 0 & 3 & 0 & 0 \\
d_x^4 & & & & & & & & & 1 & 0 & 8/3 & 1/3 & 1
\end{array} . \qquad (5.6)$$

The dual monomial of every row couples with a primal monomial in the corresponding column. From the rows we read the coefficients basis elements in $\mathcal{D}$. The leftmost 9×9 block is the identity matrix, implying the duality between $\mathcal{D}$ and $\mathcal{B}$. Then in the last four columns there are some extra monomials; these do not belong to the basis $\mathcal{B}$, yet appear with a non-zero coefficient in $\mathcal{D}$. These monomials are not in $\mathcal{Q}_\zeta$, but they can be reduced modulo $\mathcal{B}$: the last four columns yield the normal form of these monomials with respect to $\boldsymbol{\mathcal{B}}$. For example, using column 11 we find $y^3 = 3x^2y + \frac{8}{3}x^4 \pmod{\boldsymbol{\mathcal{Q}}_\zeta}$.

Using the normal form formula (5.5), we can derive the table of multiplication by x and y in the quotient algebra represented by $\boldsymbol{\mathcal{B}}$. To do this, it suffices to compute $\mathrm{NF}(y\, x^{\boldsymbol{\alpha}_i} y^{\boldsymbol{\beta}_i})$ and $\mathrm{NF}(x\, x^{\boldsymbol{\alpha}_i} y^{\boldsymbol{\beta}_i})$, for all monomials $x^{\boldsymbol{\alpha}_i} y^{\boldsymbol{\beta}_i} \in \boldsymbol{\mathcal{B}}$. This computation can be done by looking up the normal form of each monomial from the rows of matrix (5.6). The coefficients of these normal forms fill the i−th rows of the matrices

$$M_x = \begin{bmatrix} 0&1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&1/3&0&0\\ 0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{bmatrix} \quad\text{and}\quad M_y = \begin{bmatrix} 0&0&1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&1/3&0&0\\ 0&0&0&0&0&0&0&3&8/3\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&1/3\\ 0&0&0&0&0&0&0&0&0 \end{bmatrix}.$$

Computing the normal form of the border monomials of $\boldsymbol{\mathcal{B}}$ via (5.5) also yields the border basis relations and the operators of multiplication in the quotient $R/\boldsymbol{\mathcal{Q}}_\zeta$ (see e.g. [5] for more properties).

If a graded monomial ordering is fixed and $\boldsymbol{\mathcal{B}} = (\boldsymbol{x}^{\boldsymbol{\beta}_i})_{i=1,\dots,\mu}$ is the corresponding monomial basis of $R/\boldsymbol{\mathcal{Q}}_{\boldsymbol{0}}$, then $\boldsymbol{d}^{\boldsymbol{\beta}_i}$ is the leading term of (5.4) with respect to the reversed ordering (that is, we reverse the outcome of the comparison of two monomials, keeping equality unchanged) [13, Th. 3.1].

Conversely, if we are given a basis $\boldsymbol{\mathcal{D}}$ of $\mathscr{D}_\zeta$ whose coefficient matrix in the dual monomials basis $(\boldsymbol{d}^{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\notin\boldsymbol{\mathcal{Q}}_\zeta}$ is $D \in \mathbb{K}^{\mu\times s}$, we can compute a basis of $R/\boldsymbol{\mathcal{Q}}_\zeta$ by choosing μ independent columns of D, say those indexed by $\boldsymbol{d}^{\boldsymbol{\beta}_i}$, $i = 1,\dots,\mu$. If $G \in \mathbb{K}^{\mu\times\mu}$ is the (invertible) matrix formed by these columns, then $D' := G^{-1}D$, is

$$D' = \begin{array}{c} \\ \Lambda'_1 \\ \vdots \\ \Lambda'_\mu \end{array} \begin{array}{c} \begin{array}{cccccc} \boldsymbol{\beta}_1 & \cdots & \boldsymbol{\beta}_\mu & \boldsymbol{\gamma}_1 & \cdots & \boldsymbol{\gamma}_{s-\mu} \end{array} \\ \begin{bmatrix} 1 & & 0 & \lambda_{1,1} & \cdots & \lambda_{1,s-\mu} \\ & \ddots & & \vdots & & \vdots \\ 0 & & 1 & \lambda_{\mu,1} & \cdots & \lambda_{\mu,s-\mu} \end{bmatrix} \end{array}, \tag{5.7}$$

i.e. a basis of the form (5.4). Note that an arbitrary basis of $\mathscr{D}$ does not have the above diagonal form, nor does it directly provide a basis for $R/\boldsymbol{\mathcal{Q}}_\zeta$. However, a basis of this form has the desired property

$$\Lambda_i[\boldsymbol{x}^{\boldsymbol{\beta}_j}] = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases},$$

for all $i = 1,\dots,\mu$.

For $t \in \mathbb{N}$, $\mathscr{D}_t$ denotes the vector space of polynomials of $\mathscr{D}$ of degree $\leq t$. The Hilbert function $h : \mathbb{N} \to \mathbb{N}$ is defined by $h(t) = \dim(\mathscr{D}_t)$, $t \geq 0$, hence $h(0) = 1$ and $h(t) = \dim \mathscr{D}$ for $t \geq N$. The integer $h(1) - 1 = \operatorname{corank} J_f$ is known as the *breadth* of $\mathscr{D}$.

5.3 Computing Local Ring Structure

The computation of a local basis, given a system and a point, is done essentially by matrix-kernel computations, and consequently it can be carried out numerically, even when the point or even the system is inexact. Throughout the section we suppose $\boldsymbol{f} \in R^m$ and $\boldsymbol{\zeta} \in \mathbb{K}^n$ with $\boldsymbol{f}(\boldsymbol{\zeta}) = 0$.

Several matrix constructions have been proposed that use different conditions to identify the dual space as a null-space. They are based on the *stability property* of the dual basis:

$$\forall\, \Lambda \in \mathscr{D}_t, \;\; \frac{d}{d\,\partial_i}\Lambda \in \mathscr{D}_{t-1}\,, \quad i = 1, \ldots, n. \tag{5.8}$$

We list existing algorithms that compute dual-space bases:

- As pointed out in (5.2), an equivalent form of (5.8) is

$$\forall \Lambda \in \mathscr{D}_t, \;\; \Lambda[g_i\, f_i] = 0\,,\; \forall g_i \in R \quad \Longleftrightarrow \quad \Lambda[\boldsymbol{x}^{\boldsymbol{\beta}} \cdot f_i] = 0\,, \forall \boldsymbol{\beta} \in \mathbb{N}^n \tag{5.9}$$

  Macaulay's method [14] uses this equivalent characterization to derive the algorithm that is outlined in Sect. 5.3.1.
- In [17] they exploit (5.8) by forming the matrix D_i of the map

$$\frac{d}{d\,\partial_i} \;:\; \mathbb{K}[\boldsymbol{\partial}]_t \to \mathbb{K}[\boldsymbol{\partial}]_{t-1}$$

  for all $i = 1, \ldots, n$ and some triangular decomposition of the differential polynomials in terms of differential variables. This approach was used in [23] to reduce the row dimension of Macaulay's matrix, but not the column dimension.
- The closedness subspace method of Zeng [25], uses the same condition to identify a superset of $\operatorname{supp} \mathscr{D}_{t+1}$ when a basis of $\mathscr{D}_t$ is computed, and thus reduces the column dimension of the matrix.
- The *integration method* in [18] "integrates" elements of a basis of $\mathscr{D}_t$, and obtains a priori knowledge of the form of elements in degree $t + 1$ (Sect. 5.3.2).

All methods are incremental, in the sense that they start by setting $\boldsymbol{\mathcal{D}}_0 = (\mathbf{1})$ and continue by computing $\boldsymbol{\mathcal{D}}_i$, $i = 1, \ldots, N, N + 1$. When $\#\boldsymbol{\mathcal{D}}_N = \#\boldsymbol{\mathcal{D}}_{N+1}$ then $\boldsymbol{\mathcal{D}}_N$ is a basis of $\mathscr{D}$, and N is the nilindex of $\boldsymbol{\mathcal{Q}}$.

We shall review two of these approaches to compute a basis for $\mathscr{D}$, and then describe an improvement, that allows simultaneous computation of a monomial basis of the quotient ring, while avoiding redundant computations.

5.3.1 Macaulay's Dialytic Matrices

This matrix construction is presented in [14, Ch. 4], a modern introduction is contained in [4], together with an implementation of the method in ApaTools.[1]

The idea behind the algorithm is the following: An element of $\mathscr{D}$ is of the form

$$\Lambda(\boldsymbol{d}) = \sum_{|\boldsymbol{\alpha}|\leq N} \lambda_{\boldsymbol{\alpha}} \boldsymbol{d}^{\boldsymbol{\alpha}}$$

under the condition: $\Lambda^{\mathbf{0}}$ evaluates to 0 at any $g \in \langle \boldsymbol{f} \rangle$, that is,

$$\Lambda^{\mathbf{0}}(g) = \Lambda^{\mathbf{0}}\left(\sum g_i f_i\right) = 0 \iff \Lambda^{\mathbf{0}}(x^{\boldsymbol{\beta}} f_i) = 0 \ ,$$

for all monomials $x^{\boldsymbol{\beta}}$, $\boldsymbol{\beta} \in \mathbb{N}$.

If we apply this condition recursively for $|\boldsymbol{\alpha}| \leq N$, we get a vector of coefficients $(\lambda_{\boldsymbol{\alpha}})_{|\boldsymbol{\alpha}|\leq N}$ in the (right) kernel of the matrix with rows indexed by constraints $\Lambda^{\mathbf{0}}[\boldsymbol{x}^{\boldsymbol{\beta}} \boldsymbol{f}_i] = 0$, $|\boldsymbol{\beta}| \leq N-1$. A basis of $\boldsymbol{\mathcal{D}}_N$ is given by the kernel of this matrix in depth N. The method consists in computing the kernel of these matrices for $N = 1, 2, \ldots$; when N reaches the nilindex of $\boldsymbol{\mathcal{I}}$, For some value of N, this kernel stabilizes and the generating vectors form a basis of $\mathscr{D}$.

Note that the only requirement is to be able to perform derivation of the input equations and evaluation at $\boldsymbol{\zeta} = \mathbf{0}$.

Example 5.4 Let $f_1 = x_1 - x_2 + x_1^2$, $f_2 = x_1 - x_2 + x_2^2$. We also refer the reader to [4, Ex. 2] for a detailed demonstration of this instance. The matrices in order 1 and 2 are:

$$\begin{array}{c} \\ f_1 \\ f_2 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} 1 & d_1 & d_2 \end{array} \\ \begin{bmatrix} 0 & 1 & -1 \\ 0 & 1 & -1 \end{bmatrix} \end{array}, \quad \begin{array}{c|cccccc} & 1 & d_1 & d_2 & d_1^2 & d_1 d_2 & d_2^2 \\ \hline f_1 & 0 & 1 & -1 & 1 & 0 & 0 \\ f_2 & 0 & 1 & -1 & 0 & 0 & 1 \\ x_1 f_1 & 0 & 0 & 0 & 1 & -1 & 0 \\ x_1 f_2 & 0 & 0 & 0 & 1 & -1 & 0 \\ x_2 f_1 & 0 & 0 & 0 & 0 & 1 & -1 \\ x_2 f_2 & 0 & 0 & 0 & 0 & 1 & -1 \end{array}.$$

[1] http://www.neiu.edu/~zzeng/apatools.htm

The kernel of the left matrix gives $\boldsymbol{\mathcal{D}}_1 = (1, d_1 + d_2)$. Expanding up to order 2, we get the matrix on the right, and $\boldsymbol{\mathcal{D}}_2 = (1, d_1 + d_2, -d_1 + d_1^2 + d_1 d_2 + d_2^2)$. If we expand up to depth 3 we get the same null-space, thus $\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}}_2$.

5.3.2 Integration Method

This method is presented in [18]. It is an evolution of Macaulay's method, since the matrices are not indexed by all differentials, but just by elements based on knowledge of the previous step. This performs a computation adapted to the given input and results in smaller matrices.

For $\Lambda \in \mathbb{K}[\boldsymbol{\partial}]$, we denote by $\textstyle\int_k \Lambda$ the element $\Phi \in \mathbb{K}[\boldsymbol{\partial}]$ with the property $\frac{d}{d\partial_k}\Phi(\boldsymbol{\partial}) = \Lambda(\boldsymbol{\partial})$ and with no constant term with respect to ∂_k.

Theorem 5.3 ([18, Th. 15]) *Let $\{\Lambda_1, \Lambda_2, \dots, \Lambda_s\}$ be a basis of $\mathscr{D}_{t-1}$, that is, the subspace of $\mathscr{D}$ of elements of order at most $t-1$. An element $\Lambda \in \mathbb{K}[\boldsymbol{\partial}]$ with no constant term lies in $\mathscr{D}_t$ iff it is of the form:*

$$\Lambda(\boldsymbol{\partial}) = \sum_{i=1}^{s}\sum_{k=1}^{n} \lambda_{ik} \textstyle\int_k \Lambda_i(\partial_1, \dots, \partial_k, 0, \dots, 0), \tag{5.10}$$

for $\lambda_{ik} \in \mathbb{K}$, and the following two conditions hold:

(i) $\displaystyle\sum_{i=1}^{s} \lambda_{ik} \frac{d}{d\partial_l}\Lambda_i(\boldsymbol{\partial}) - \sum_{i=1}^{s} \lambda_{il} \frac{d}{d\partial_k}\Lambda_i(\boldsymbol{\partial}) = 0,$ *for all* $1 \le k < l \le n$.

(ii) $\Lambda^{\zeta}[f_k] = 0,$ *for* $k = 1, \dots, m$.

Condition (i) is equivalent to $\frac{d}{d\partial_k}\Lambda \in \mathscr{D}_{t-1}$, for all k. Thus the two conditions express exactly the fact that $\mathscr{D}$ must be stable under derivation and its members must vanish on $\langle \boldsymbol{f} \rangle$.

This gives the following algorithm to compute the dual basis: Start with $\boldsymbol{\mathcal{D}}_0 = \langle 1 \rangle$. Given a basis of $\mathscr{D}_{t-1}$ we generate the ns candidate elements $\textstyle\int_k \Lambda_{i-1}(\partial_1, \dots, \partial_k, 0, \dots, 0)$. Conditions (i) and (ii) give a linear system with unknowns λ_{ik}. The columns of the corresponding matrix are indexed by the candidate elements. Then, the kernel of this matrix gives a basis of $\mathscr{D}_t$, which we use to generate new candidate elements. If for some t we compute a kernel of the same dimension as $\mathscr{D}_{t-1}$, then we have a basis of $\mathscr{D}$.

Example 5.5 Consider the instance of Example 5.4, $f_1 = x_1 - x_2 + x_1^2$, $f_2 = x_1 - x_2 + x_2^2$. We have $f_1(\boldsymbol{\zeta}) = f_2(\boldsymbol{\zeta}) = 0$, thus we set $\boldsymbol{\mathcal{D}}_0 = \{1\}$. Equation (5.10) gives $\Lambda = \lambda_1 d_1 + \lambda_2 d_2$. Condition (i) induces no constraints and (ii) yields the system

$$\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = 0 \tag{5.11}$$

where the columns are indexed by d_1, d_2. We get $\lambda_1 = \lambda_2 = 1$ from the kernel of this matrix, thus $\mathcal{D}_1 = \{1, d_1 + d_2\}$.

For the second step, we compute the elements of $\mathcal{D}_2$, that must be of the form

$$\Lambda = \lambda_1 d_1 + \lambda_2 d_2 + \lambda_3 d_1^2 + \lambda_4 (d_1 d_2 + d_2^2).$$

Condition (i) yields $\lambda_3 - \lambda_4 = 0$, and together with (ii) we form the system

$$\begin{bmatrix} 0 & 0 & 1 & -1 \\ 1 & -1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_4 \end{bmatrix} = 0, \tag{5.12}$$

with columns indexed by $d_1, d_2, d_1^2, d_1 d_2 + d_2^2$. We get two vectors in the kernel, the first yielding again $d_1 + d_2$ and a second one for $\lambda_1 = -1, \lambda_2 = 0, \lambda_3 = \lambda_4 = 1$, so we deduce that $-d_1 + d_1^2 + d_1 d_2 + d_2^2$ is a new element of $\mathcal{D}_2$.

In the third step we have

$$\begin{aligned} \Lambda = \quad & \lambda_1 d_1 + \lambda_2 d_2 + \lambda_3 d_1^2 + \lambda_4 (d_1 d_2 + d_2^2) + \\ & \lambda_5 (d_1^3 - d_1^2) + \lambda_6 (d_2^3 + d_1 d_2^2 + d_1^2 d_2 - d_1 d_2), \end{aligned} \tag{5.13}$$

condition (i) leads to $\lambda_3 - \lambda_4 + (\lambda_5 - \lambda_6)(d_1 + d_2) = 0$, and together with condition (ii) we arrive at

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & -1 & 1 & 0 & -1 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_6 \end{bmatrix} = 0, \tag{5.14}$$

of size 4 × 6, having two kernel elements that are already in $\mathcal{D}_2$. We derive that $\mathscr{D} = \langle \mathcal{D}_2 \rangle = \langle \mathcal{D}_3 \rangle$ and the algorithm terminates.

Note that for this example Macaulay's method ends with a matrix of size 12 × 10, instead of 4 × 6 in this approach.

5.3.3 Computing a Primal-Dual Pair

In this section we provide a process that allows simultaneous computation of a basis pair $(\mathcal{D}, \mathcal{B})$ of $\mathscr{D}$ and $R/\mathcal{Q}$.

Computing a basis of $\mathscr{D}$ degree by degree involves duplicated computations. The successive spaces computed are $\mathscr{D}_1 \subset \cdots \subset \mathscr{D}_N = \mathscr{D}_{N+1}$. It is more efficient to produce only new elements $\Lambda \in \mathscr{D}_t$, independent in $\mathscr{D}_t / \mathscr{D}_{t-1}$, at step t.

Also, once a dual basis is computed, one has to transform it to the form (5.4), in order to identify a basis of $R/\boldsymbol{Q}$ as well. This transformation can be done a posteriori, by finding a sub-matrix of full rank and then performing Gauss–Jordan elimination over this sub-matrix, to reach matrix form (5.7).

We introduce a condition (iii) extending Theorem 5.3, that addresses these two issues: It allows the computation of a total of μ independent elements throughout execution, and returns a "triangular" basis, e.g. a basis of $R/\boldsymbol{Q}$ is identified.

Lemma 5.2 *Let $\boldsymbol{\mathcal{D}}_{t-1} = (\Lambda_1, \ldots, \Lambda_k)$ be a basis of $\mathscr{D}_{t-1}$, whose coefficient matrix is*

$$\begin{array}{c} \\ \Lambda_1 \\ \vdots \\ \Lambda_k \end{array} \begin{array}{c} \begin{array}{cccccc} \boldsymbol{\beta}_1 & \cdots & \boldsymbol{\beta}_k & \gamma_1 & \cdots & \gamma_{s-k} \end{array} \\ \begin{bmatrix} 1 & * & * & * & \cdots & * \\ 0 & \ddots & * & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & * & \cdots & * \end{bmatrix} \end{array}, \tag{5.15}$$

yielding the monomial basis $\boldsymbol{\mathcal{B}}_{t-1} = (\boldsymbol{x}^{\boldsymbol{\beta}_i})_{i=1,\ldots,k}$. Also, let $\Lambda \in \mathbb{K}[\boldsymbol{\partial}]$ be of the form (5.10), satisfying (i–ii) of Theorem 5.3.

If we impose the additional condition:

(iii) $\Lambda^{\zeta}[\boldsymbol{x}^{\boldsymbol{\beta}_i}] = 0,\ 1 \le i \le k,$

then the kernel of the matrix implied by (i–iii) is isomorphic to $\mathscr{D}_t / \mathscr{D}_{t-1}$. Consequently, it extends $\boldsymbol{\mathcal{D}}_{t-1}$ to a basis of $\mathscr{D}_t$.

Proof Let S be the kernel of the matrix implied by (i–iii), and let $\Lambda \in \mathbb{K}[\boldsymbol{\partial}]$ be a non-zero functional in S. We have $\Lambda \in \mathscr{D}_t$ and $\Lambda^{\zeta}[\boldsymbol{x}^{\boldsymbol{\beta}_i}] = 0$ for $i = 1, \ldots, k$.

First we show that $\Lambda \notin \mathscr{D}_{t-1}$. If $\Lambda \in \mathscr{D}_{t-1}$, then $\Lambda = \sum_{i=1}^{k} \lambda_i \Lambda_i$. Take for i_0 the minimal i such that $\lambda_i \neq 0$. Then $\Lambda^{\zeta}[\boldsymbol{x}^{\boldsymbol{\beta}_{i_0}}] = \lambda_{i_0}$, which contradicts condition (iii). Therefore, $S \cap \mathscr{D}_{t-1} = \{0\}$, and S can be naturally embedded in $\mathscr{D}_t / \mathscr{D}_{t-1}$, i.e. $\dim S \le \dim \mathscr{D}_t - \dim \mathscr{D}_{t-1}$.

It remains to show that $\dim S$ is exactly $\dim \mathscr{D}_t - \dim \mathscr{D}_{t-1}$. This is true, since with condition (iii) we added $k = \dim \mathscr{D}_{t-1}$ equations, thus we excluded from the initial kernel (equal to $\mathscr{D}_t$) of (i–ii) a subspace of dimension at most $k = \dim \mathscr{D}_{t-1}$, so that $\dim S \ge \dim \mathscr{D}_t - \dim \mathscr{D}_{t-1}$.

We deduce that $S \cong \mathscr{D}_t / \mathscr{D}_{t-1}$, thus a basis of S extends $\boldsymbol{\mathcal{D}}_{t-1}$ to a basis of $\mathscr{D}_t$.

□

The above condition is easy to realize; it is equivalent to $\forall i$, $\boldsymbol{d}^{\boldsymbol{\beta}_i} \notin \operatorname{supp} \Lambda$, which implies adding a row (linear constraint) for every i. If we choose the elements of $\boldsymbol{\mathcal{B}}$ with a "reversed" total degree ordering (if a monomial compares total-degree "less than" another one, then it compares "bigger than" the same monomial in the reversed order), then in many cases this constraint becomes $\lambda_{ik} = 0$ for some i, k. In this case we rather remove the column corresponding to λ_{ik} instead of adding a row.

Hence this lemma allows to shrink the kernel (but also the dimension) of the matrix and compute only new dual elements, which are reduced modulo the previous basis. For a detailed size comparison, see Table 5.1.

Let us explore our running example, to demonstrate the essence of this improvement.

Example 5.6 We re-run Example 5.5 using Lemma 5.2. In the initialization step $\mathcal{D}_0 = (1)$ is already in triangular form with respect to $\mathcal{B}_0 = \{1\}$. For the first step, we demand $\Lambda[1] = 0$, thus the matrix is the same as (5.11), yielding $\mathcal{D}_1 = (1, d_1 + d_2)$. We extend $\mathcal{B}_1 = \{1, x_2\}$, so that $\mathcal{D}_1$ is triangular with respect to $\mathcal{B}_1$.

In the second step we remove from (5.12) the second column, hence we are left with the 3 × 3 system

$$\begin{bmatrix} 0 & 1 & -1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = 0,$$

yielding a single solution $-d_1 + d_1^2 + d_1 d_2 + d_2^2$. We extend $\mathcal{B}_1$ by adding the monomial x_1: $\mathcal{B}_2 = \{1, x_2, x_1\}$.

For the final step, we search an element of the form (5.13) with $\Lambda[x_1] = \Lambda[x_2] = 0$, and together with (i–ii) we get:

$$\begin{bmatrix} 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_3 \\ \vdots \\ \lambda_6 \end{bmatrix} = 0.$$

We find an empty kernel, thus we recover the triangular basis $\mathcal{D} = \mathcal{D}_2$, which can be diagonalized to reach the form:

$$\begin{array}{c|cccccc|} & 1 & d_2 & d_1 & d_1^2 & d_1 d_2 & d_2^2 \\ \Lambda_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \Lambda_2 & 0 & 1 & 0 & 1 & 1 & 1 \\ \Lambda_3 & 0 & 0 & 1 & -1 & -1 & -1 \end{array}.$$

This diagonal basis is dual to the basis $\mathcal{B} = (1, x_2, x_1)$ of the quotient ring and also provides a normal form algorithm (Lemma 5.1) with respect to $\mathcal{B}$. In the final step we generated a 4 × 4 matrix, of smaller size compared to all previous methods.

Another example is treated in Fig. 5.1, with the aid of pictures.

This technique for computing $\mathcal{B}$ can be applied similarly to other matrix methods, e.g. Macaulay's dialytic method.

If $h(t) - h(t-1) > 1$, i.e. there is more than one element in step t, then the choice of monomials to add to $\mathcal{B}$ is obtained by extracting a non-zero maximal

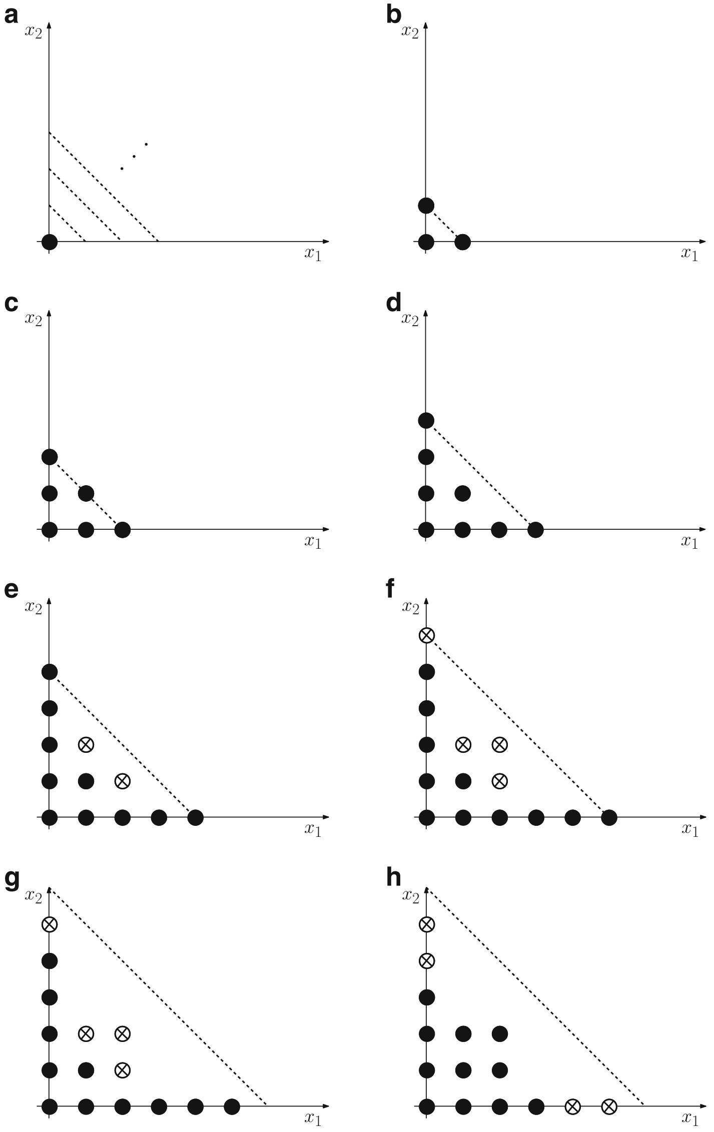


Fig. 5.1 Discovering a primal-dual basis pair for the root $\boldsymbol{\zeta} = (0,0)$ of the bivariate system $\{2x_1x_2^2 + 5x_1^4,\ 2x_1^2x_2 + 5x_2^4\}$. (**a**) $\boldsymbol{\zeta} = (0,0)$ nullifies the system, so monomial 1 is inserted in $\boldsymbol{\mathcal{B}}$. (**b**) In degree 1 we have d_1 and d_2, so x_1 and x_2 are added to $\boldsymbol{\mathcal{B}}$. (**c**) In degree 2, we add the dual monomials of d_1^2, d_1d_2, and d_2^2 to $\boldsymbol{\mathcal{B}}$. (**d**) In degree 3, two dual basis elements appear d_1^3, d_2^3. (**e**) In degree 4 we get $2d_1^4 - 5d_1d_2^2$, $2d_2^4 - 5\,d_1^2d_2$, and we choose x_1^4. (**f**) In degree 5, we get $2d_1^5 + 2d_2^5 - 5\,d_1^2d_2^2$ therefore x_1^5 is a primal monomial. (**g**) We find no element of degree 6 and the algorithm stops: $\boldsymbol{\zeta}$ has multiplicity 11. (**h**) Different choices at each step lead to different bases; here is another possible result

minor from the coefficient matrix in $(\boldsymbol{d}^{\alpha})$. In practice, we will look first at the minimum monomials with respect to a fixed term ordering.

5.4 Deflation of a Singular Point

Deflation techniques allow to transform a system of equations defining a singular solution into a new system where the solution corresponds to a simple point. Usually this is done by adding new variables and new equations so that a simple isolated solution of the extended system projects onto the singular solution of the initial system. We will illustrate different types of deflation.

5.4.1 *The Univariate Case*

In preparation for the multivariate case, we review an approach for treating singularities of univariate polynomials.

Let $g(x) \in \mathbb{K}[x]$ be a polynomial which attains at $x = 0$ a root of multiplicity $\mu > 1$. The latter is defined as the positive integer μ such that $d^{\mu} g(0) \neq 0$ whereas $g(0) = dg(0) = \cdots = d^{\mu-1} g(0) = 0$. Here we denote by $d^k g(x) = \dfrac{1}{k!}\dfrac{d^k}{dx^k} g(x)$ the normalized derivative of order k with respect to x.

We see that $\mathscr{D}_0 = \langle 1, d, \ldots, d^{\mu-1} \rangle$ is the maximal space of differentials which is stable under derivation, that vanish when applied to members of $\mathcal{Q}_0$, the $\langle x \rangle$−primary component of $\langle g \rangle$ at $x = 0$.

Example 5.7 Let $g(x) = (x-1)^4$, and $\zeta = 1$. First we check that the space of differentials that vanish on the solution include all linear combinations of $\mathscr{D}_{\zeta} = \langle 1, d_{\zeta}, d_{\zeta}^2, d_{\zeta}^3 \rangle$. For instance, we compute $d^2[g] = 6(x-1)^2|_{\zeta} = 0$.

Now d^4 is not a member of $\mathscr{D}_{\zeta}$ since $d^4[g] = 1 \neq 0$ does not vanish. Similarly, for all $i \geq 4$, there exists a member of the ideal generated by g which does not evaluate to zero when we apply the differential d^i, namely $d^i[x^{i-4} g] = 1$.

We conclude that the local dual space is exactly $\mathscr{D}_{\zeta}$, and verify that ζ 4−fold zero of f.

Consider now the symbolically perturbed equation

$$f_1(x, \boldsymbol{\varepsilon}) = g(x) + \varepsilon_1 + \varepsilon_2 x + \cdots + \varepsilon_{\mu-1} x^{\mu-2} \tag{5.16}$$

and apply every basis element of $\mathscr{D}_0$ to arrive at the new system

$$\boldsymbol{f}(x, \boldsymbol{\varepsilon}) = \left(f_1, df_1, \ldots, d^{\mu-1} f_1\right)$$

in $\mu - 1$ variables. The i–th equation has the form

$$f_i = d^{i-1} f_1 = d^{i-1} g + \sum_{k=i}^{\mu-1} \binom{k-1}{i-1} x^{k-i} \varepsilon_k \ ,$$

i.e linear in $\boldsymbol{\varepsilon}$, the last one being $f_\mu = d^{\mu-1} g(x)$. This system deflates the root, since the determinant of its Jacobian matrix at $(0, \mathbf{0})$ is

$$\det J_{\boldsymbol{f}}(0,\mathbf{0}) = \left|\begin{array}{c|cc} \frac{d}{dx} f_1 & 1 & 0 \\ \vdots & & \ddots \\ \frac{d}{dx} f_{\mu-1} & 0 & 1 \\ \hline \frac{d}{dx} f_\mu & \multicolumn{2}{c}{0} \end{array}\right| \begin{array}{l} = -df_\mu(0) \\ = -\mu d^\mu g(0) \neq 0. \end{array}$$

Now suppose that ζ^* is an approximate zero, close to $x = \zeta$. We can still compute $\mathcal{D}_\zeta$ by evaluating $g(x)$ and the derivatives up to a threshold relative to the error present in ζ^*. Then we can form (5.16) and use verification techniques to certify the root. Checking that the Newton operator is contracting shows the existence and unicity of a multiple root in a neighborhood of the input data. We are going to extend this approach, described in [22], to multi-dimensional isolated multiple roots.

5.4.2 Deflation Using the Dialytic Approach

Let us consider a system of equations $\boldsymbol{f} = (f_1, \ldots, f_s)$, $f_k \in \mathbb{R}[\boldsymbol{x}]$, $s \geq n$, which has an isolated zero $\boldsymbol{\zeta}$.

If the Jacobian matrix

$$J_{\boldsymbol{f}}(\boldsymbol{x}) = \begin{pmatrix} \partial_{x_1} f_1(\boldsymbol{x}) \cdots \partial_{x_n} f_1(\boldsymbol{x}) \\ \vdots \qquad\qquad \vdots \\ \partial_{x_1} f_s(\boldsymbol{x}) \cdots \partial_{x_n} f_s(\boldsymbol{x}) \end{pmatrix}$$

at the point $\boldsymbol{\zeta}$ is of (maximal) rank n, then the root $\boldsymbol{\zeta}$ is simple. Moreover the iteration

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + J_{\boldsymbol{f}}(\boldsymbol{x}^{(n)})^{+} \boldsymbol{f}(\boldsymbol{x}^{(n)})$$

converges to $\boldsymbol{\zeta}$ as soon as the initial point $\boldsymbol{x}^{(0)}$ is close enough to $\boldsymbol{\zeta}$ [9].

When the root is not simple, then the rank of $J_{\boldsymbol{f}}(\boldsymbol{\zeta})$ is $r_1 < n$ and there are $n - r_1$ linearly independent differentials of order 1 of the form $u_1 \partial_{x_1} + \cdots + u_n \partial_{x_n}$ which satisfy

$$u_1 \partial_{x_1} \mathbf{f}(\boldsymbol{\zeta}) + \cdots + u_n \partial_{x_n} \mathbf{f}(\boldsymbol{\zeta}) = 0, \tag{5.17}$$

or equivalently

$$J_f(\boldsymbol{\zeta})\begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} = \mathbf{0}.$$

To fix a solution of this system, we can choose $n - r_1$ random vectors $\boldsymbol{r}_j = (r_{0,j}, r_{1,j}, \ldots, r_{j,n})$ for $j = 1, \ldots, n - r_1$ and consider the equations

$$u_1 r_{1,j} + \cdots + u_n r_{j,n} + r_{j,0} = 0, \quad j = 1, \ldots, n - r_1. \tag{5.18}$$

Extending the initial system of equations $\boldsymbol{f}(\boldsymbol{x})$ with the new Eqs. (5.17) and (5.18), we obtain a new system of equations denoted $\boldsymbol{f}_{[1]}(\boldsymbol{x}_{[1]})$ in the variables $\boldsymbol{x}_{[1]} = (x_1, \ldots, x_n, u_1, \ldots, u_n)$. This system is called a system *deflated* from $\boldsymbol{f}$.

By construction, if $\boldsymbol{\zeta}$ is an isolated root of $\boldsymbol{f}(\boldsymbol{x}) = 0$ and rank $J_f(\boldsymbol{\zeta}) = r_1$, there is a unique $\boldsymbol{u}^*$ satisfying Eqs. (5.17) and (5.18). Thus $\boldsymbol{x}^*_{[1]} = (\boldsymbol{\zeta}, \boldsymbol{u}^*)$ is an isolated solution of the system $\boldsymbol{f}_{[1]}(\boldsymbol{x}_{[1]}) = 0$.

If the root $\boldsymbol{x}^*_{[1]}$ of this system is simple, then Newton iteration applied on $\boldsymbol{f}_{[1]}$ will converge quadratically to $\boldsymbol{x}^*_{[1]}$ for an initial point $\boldsymbol{x}^{(0)}_{[1]}$ in its neighborhood.

If the root is not simple, the deflation can be applied to the system $\boldsymbol{f}_{[1]}(\boldsymbol{x}_{[1]}) = 0$ and we obtain a new system $\boldsymbol{f}_{[2]}(\boldsymbol{x}_{[2]}) = 0$ in $4\,n$ variables.

As shown in the next result, the process can be applied inductively until the root becomes simple:

Theorem 5.4 ([3, 12]) *If $\boldsymbol{\zeta}$ is an isolated root of the system $\boldsymbol{f}(\boldsymbol{x}) = 0$, there exists a number $k \in \mathbb{N}$ such that $\boldsymbol{f}_{[k]}(\boldsymbol{x}_{[k]}) = 0$ has a simple root $\boldsymbol{x}^*_{[k]}$ whose projection on the first n coordinates is $\boldsymbol{\zeta}$.*

It is proved in [12] (or in [3]) that the number k of iterations is at most the depth of the multiplicity of $\boldsymbol{f}$ at $\boldsymbol{\zeta}$, that is the maximum degree of a differential polynomial of the inverse system of $\boldsymbol{f}$ at $\boldsymbol{\zeta}$.

Notice that the number of variables of the system $\boldsymbol{f}_{[k]}$ is $n \times 2^k$.

Example 5.8 Consider the system $f_1(x_1, x_2) = 2x_1x_2^2 + 5x_1^4$, $f_2(x_1, x_2) = 2x_1^2x_2 + 5x_2^4$ and the singular point $\boldsymbol{\zeta} = (0, 0)$. Since $J_f(\boldsymbol{\zeta}) = 0$, we apply a first deflation step, i.e. we compute the equations $J_f(\boldsymbol{x})\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$, and two random linear equations:

$$\begin{aligned}
&g_1(\boldsymbol{x}_{[1]}) = f_1 = 2\,x_1{x_2}^2 + 5\,{x_1}^4, \;\; g_2(\boldsymbol{x}_{[1]}) = f_2 = 2\,{x_1}^2x_2 + 5\,{x_2}^4 \\
&g_3(\boldsymbol{x}_{[1]}) \quad = \quad \left(2\,{x_2}^2 + 20\,{x_1}^3\right)u_1 + 4\,x_1x_2u_2 \\
&g_4(\boldsymbol{x}_{[1]}) \quad = \quad 4\,x_1x_2u_1 + \left(2\,{x_1}^2 + 20\,{x_2}^3\right)u_2 \\
&g_5(\boldsymbol{x}_{[1]}) \quad = \quad 16\,u_1 + u_2 - 1, \quad g_6(\boldsymbol{x}_{[1]}) = 70\,u_1 + 77\,u_2
\end{aligned}$$

The new Jacobian matrix $J_{\boldsymbol{g}}(\boldsymbol{\zeta}_{[1]})$ is rank-defect, with $\boldsymbol{\zeta}_{[1]} = (\boldsymbol{x},\boldsymbol{u}) = \left(0, 0, \frac{11}{166}, \frac{-5}{83}\right)$ is zero, and the multiplicity has dropped from 11 to 4. Therefore we repeat the procedure for this new system using random equations from the kernel of $J_{(\boldsymbol{g})}(\boldsymbol{\zeta}_{[1]})$ (5.18):

$$
\begin{aligned}
h_1(\boldsymbol{x}_{[2]}) &= 2\,x_1{x_2}^2 + 5\,{x_1}^4,\ \ h_2(\boldsymbol{x}_{[2]}) = 2\,{x_1}^2x_2 + 5\,{x_2}^4\\
h_3(\boldsymbol{x}_{[2]}) &= \left(2\,{x_2}^2 + 20\,{x_1}^3\right)u_1 + 4\,x_1x_2u_2\\
h_4(\boldsymbol{x}_{[2]}) &= 4\,x_1x_2u_1 + \left(2\,{x_1}^2 + 20\,{x_2}^3\right)u_2\\
h_5(\boldsymbol{x}_{[2]}) &= 16\,u_1 + u_2 - 1,\ \ \ h_6(\boldsymbol{x}_{[2]}) = 70\,u_1 + 77\,u_2\\
h_7(\boldsymbol{x}_{[2]}) &= \left(2\,{x_2}^2{+}20\,{x_1}^3\right)v_1{+}4\,x_1x_2v_2,\ \ h_8(\boldsymbol{x}_{[2]}) = 4\,x_1x_2v_1{+}\left(2\,{x_1}^2{+}20\,{x_2}^3\right)v_2\\
h_9(\boldsymbol{x}_{[2]}) &= \left(60\,{x_1}^2u_1{+}4\,x_2u_2\right)v_1{+}\left(4\,x_2u_1{+}4\,x_1u_2\right)v_2{+}\left(2\,{x_2}^2{+}20\,{x_1}^3\right)v_3{+}4\,x_1x_2v_4\\
h_{10}(\boldsymbol{x}_{[2]}) &= \left(4\,x_2u_1{+}4\,x_1u_2\right)v_1{+}\left(4\,x_1u_1{+}60\,{x_2}^2u_2\right)v_2{+}4\,x_1x_2v_3{+}\left(2\,{x_1}^2{+}20\,{x_2}^3\right)v_4\\
h_{11}(\boldsymbol{x}_{[2]}) &= 16\,v_3{+}v_4,\ \ h_{12}(\boldsymbol{x}_{[2]}) = 70\,v_3{+}77\,v_4\\
h_{13}(\boldsymbol{x}_{[2]}) &= 53\,v_1{+}12\,v_2{+}19\,v_3{+}63\,v_4{-}1,\ \ h_{14}(\boldsymbol{x}_{[2]}) = 40\,v_1{+}90\,v_2{+}3\,v_3{+}49\,v_4
\end{aligned}
$$

We obtain a new system which has a regular root at $\boldsymbol{\zeta}_{[2]} = (\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}) = \left(0, 0, \frac{11}{166}, \frac{-5}{83}, \frac{3}{143}, \frac{-4}{429}, 0, 0\right)$, so deflation is achieved in two steps. We see that the number of (equations and) variables increased exponentially, from 2 to 8, and the system is no longer square.

5.4.3 *Deflation Using the Inverse System*

We consider again a system of equations $\boldsymbol{f} = (f_1, \ldots, f_s)$, $f_k \in \mathbb{R}[\boldsymbol{x}]$, which has an isolated root $\boldsymbol{\zeta}$ of multiplicity μ.

In this section, we will also extend the initial system by introducing new variables so that the extended system has a simple isolated root, which projects onto the multiple point $\boldsymbol{\zeta}$. Contrarily to the deflation technique described in Sect. 5.4.2, the number of new variables will be directly related with the multiplicity μ of the point. Let $\boldsymbol{b} = ((\boldsymbol{x}-\boldsymbol{\zeta})^{\boldsymbol{\beta}_1}, \ldots, (\boldsymbol{x}-\boldsymbol{\zeta})^{\boldsymbol{\beta}_\mu})$ be a basis of $R/\mathcal{Q}_\zeta$ and $\boldsymbol{\mathcal{D}} = (\Lambda_1, \ldots, \Lambda_\mu)$ its dual counterpart, with $\beta_1 = (0, \ldots, 0)$, $\Lambda_1 = \mathbf{1}$.

We introduce a new set of equations starting from $\boldsymbol{f}$, as follows: add to every f_k the polynomial $g_k = f_k + p_k$, $p_k = \sum_{i=1}^{\mu} \varepsilon_{i,k}(\boldsymbol{x}-\boldsymbol{\zeta})^{\boldsymbol{\beta}_i}$ where $\boldsymbol{\varepsilon}_k = (\varepsilon_{k,1}, \ldots, \varepsilon_{k,\mu})$ is a new vector of μ variables.

Consider the system

$$\boldsymbol{\mathcal{D}g}(\boldsymbol{x},\boldsymbol{\varepsilon}) = \Big(\Lambda_1(\boldsymbol{\partial_x})[\boldsymbol{g}],\ldots,\Lambda_\mu(\boldsymbol{\partial_x})[\boldsymbol{g}]\Big).$$

where $\Lambda^x[g_k] = \Lambda_i(\boldsymbol{d_x})[g_k]$ is defined as in (5.1) with $\boldsymbol{\zeta}$ replaced by $\boldsymbol{x}$, i.e. we differentiate g_k but we do not evaluate at $\boldsymbol{\zeta}$. This is a system of μs equations, which we shall index $\boldsymbol{\mathcal{D}g}(\boldsymbol{x},\boldsymbol{\varepsilon}) = (g_{1,1},\ldots,g_{\mu,s})$. We have

$$g_{ik}(\boldsymbol{x},\boldsymbol{\varepsilon}) = \Lambda_i^x[f_k + p_k] = \Lambda_i^x[f_k] + \Lambda_i^x[p_k] = \Lambda_i^x[f_k] + p_{i,k}(\boldsymbol{x},\boldsymbol{\varepsilon}).$$

Notice that $p_{i,k}(\boldsymbol{\zeta},\boldsymbol{\varepsilon}) = \Lambda_i^{\zeta}[p_k] = \varepsilon_{i,k}$ because $\boldsymbol{\mathcal{D}} = (\Lambda_1,\ldots,\Lambda_\mu)$ is dual to $\boldsymbol{b}$.

As the first basis element of $\boldsymbol{\mathcal{D}}$ is $\mathbf{1}$ (the evaluation at the root), the first s equations are $\boldsymbol{g}(\boldsymbol{x},\boldsymbol{\varepsilon}) = 0$.

Note that this system is under-determined, since the number of variables is $\mu\, s+n$ and the number of equations is μs. We shall provide a systematic way to choose n variables and purge them (or better, set them equal to zero).

By linearity of the Jacobian matrix we have

$$\begin{aligned} J_{\boldsymbol{\mathcal{D}g}}(\boldsymbol{x},\boldsymbol{\varepsilon}) &= J_{\boldsymbol{\mathcal{D}f}}(\boldsymbol{x},\boldsymbol{\varepsilon}) + J_{\boldsymbol{\mathcal{D}p}}(\boldsymbol{x},\boldsymbol{\varepsilon}) \\ &= [\, J_{\boldsymbol{\mathcal{D}f}}(\boldsymbol{x}) \,|\, \mathbf{0}\ ] + [\, J^x_{\boldsymbol{\mathcal{D}p}}(\boldsymbol{x},\boldsymbol{\varepsilon}) \,|\, J^{\varepsilon}_{\boldsymbol{\mathcal{D}p}}(\boldsymbol{x},\boldsymbol{\varepsilon})\,], \end{aligned} \tag{5.19}$$

where $J^x_{\boldsymbol{\mathcal{D}p}}(\boldsymbol{x},\boldsymbol{\varepsilon})$ (resp. $J^{\varepsilon}_{\boldsymbol{\mathcal{D}p}}(\boldsymbol{x},\boldsymbol{\varepsilon})$) is the Jacobian matrix of $\boldsymbol{\mathcal{D}p}$ with respect to $\boldsymbol{x}$ (resp. $\boldsymbol{\varepsilon}$). By construction the Jacobian matrix $J^{\varepsilon}_{\boldsymbol{\mathcal{D}p}}(\boldsymbol{x},\boldsymbol{\varepsilon})$ of the system $\boldsymbol{p} = (\Lambda_i^x(p_j))_{1\le i,j\le\mu}$ is, up to a reordering of the rows and columns, a block diagonal matrix with s blocks of the form

$$\left(\Lambda_i^x[\boldsymbol{b}_j]\right)_{1\le i,j\le\mu}.$$

As $\boldsymbol{\mathcal{D}}$ is dual to the basis $\boldsymbol{b}$, $\left(\Lambda_i^x[\boldsymbol{b}_j](\boldsymbol{\zeta},\mathbf{0})\right)_{1\le i,j\le\mu}$ is the identity matrix, the Jacobian $J^{\varepsilon}_{\boldsymbol{\mathcal{D}p}}(\boldsymbol{x},\boldsymbol{\varepsilon})$ evaluated at $(\boldsymbol{\zeta},\mathbf{0})$ is, up to a reordering of the crows and columns, the identity matrix of dimension μs.

Using decomposition (5.19), we easily deduce the following property:

Lemma 5.3 *The $\mu\, s \times \mu\, s$ Jacobian matrix $J^{\varepsilon}_{\boldsymbol{\mathcal{D}g}}(\boldsymbol{x},\boldsymbol{\varepsilon})$ is of full rank $\mu\, s$ at $(\boldsymbol{\zeta},\mathbf{0})$.*

Another interesting property is the following [15]:

Lemma 5.4 *The $\mu s \times n$ Jacobian matrices $J^x_{\boldsymbol{\mathcal{D}g}}(\boldsymbol{x},\boldsymbol{\varepsilon})$ and $J^x_{\boldsymbol{\mathcal{D}f}}(\boldsymbol{x},\boldsymbol{\varepsilon})$ are of full rank n at $(\boldsymbol{\zeta},\mathbf{0})$.*

We are going to use these properties to construct sub-systems of $\boldsymbol{\mathcal{D}g}$ with a simple root "above" $\boldsymbol{\zeta}$.

The columns of $J_{\boldsymbol{\mathcal{D}g}}(\boldsymbol{x},\boldsymbol{\varepsilon})$ are indexed by the variables $(\boldsymbol{x},\boldsymbol{\varepsilon})$, while the rows are indexed by the polynomials g_{ik}. We construct the following systems:

(a) Let $\boldsymbol{\mathcal{D}f}^I$ be a subsystem of $\boldsymbol{\mathcal{D}f}$ s.t. the corresponding n rows of $J_{\boldsymbol{\mathcal{D}f}}(\boldsymbol{\zeta})$ are linearly independent (Lemma 5.4 implies that such a subset exists). We denote by $I = \{(i_1,k_1),\ldots,(i_n,k_n)\}$ their indices.

(b) Let $\boldsymbol{\mathcal{D}}\tilde{\boldsymbol{g}}(\boldsymbol{x}, \tilde{\boldsymbol{\varepsilon}})$ be the square system formed by removing the variables $\varepsilon_{k_1,i_1}, \dots, \varepsilon_{k_n,i_n}$ from $\boldsymbol{\mathcal{D}}\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\varepsilon})$. Therefore the Jacobian $J_{\boldsymbol{\mathcal{D}}\tilde{\boldsymbol{g}}}(\boldsymbol{x}, \tilde{\boldsymbol{\varepsilon}})$ derives from $J_{\boldsymbol{\mathcal{D}}\boldsymbol{g}}(\boldsymbol{x}, \boldsymbol{\varepsilon})$, after purging the columns indexed by $\varepsilon_{k_1,i_1}, \dots, \varepsilon_{k_n,i_n}$ and it's (i_j, k_j)-th row becomes $[\nabla(\Lambda^{\boldsymbol{x}}_{i_j}\tilde{g}_{i_j,k_j})^T \,|\, \mathbf{0}\,]$.

A first consequence is the following result, giving a $n \times n$ system deduce from the initial system $\boldsymbol{f}$, with a simple root at $\boldsymbol{\zeta}$:

Theorem 5.5 (Deflation Theorem 1 [15]) *Let $\boldsymbol{f}(\boldsymbol{x})$ be a $n-$variate polynomial system with an $\mu-$fold isolated zero at $\boldsymbol{x} = \boldsymbol{\zeta}$. Then the $n \times n$ system $\boldsymbol{\mathcal{D}}\boldsymbol{f}^I(\boldsymbol{x}) = 0$, defined in (a), has a simple root at $\boldsymbol{x} = \boldsymbol{\zeta}$.*

Example 5.9 In our running example, we expand the rectangular Jacobian matrix of 6 polynomials in (x_1, x_2). Choosing the rows corresponding to f_1 and $(d_1 - d_2^2 - d_1d_2 - d_1^2)[f_1]$, we find a non-singular minor, hence the resulting system $(f_1, 2x_1)$ has a regular root at $\boldsymbol{\zeta} = (0, 0)$.

The deflated system $\boldsymbol{\mathcal{D}}\boldsymbol{f}^I(\boldsymbol{x}) = 0$ is a square system in n variables. Contrarily to the deflation approach in [4, 12], we do not introduce new variables and one step of deflation is provably sufficient. The trade-off is that here we assume that exact dual elements are pointed at by indices I, so as to be able to compute the original multiple root with high accuracy.

On the other hand, when the coefficients are machine numbers, an exact multiple root is unlikely to exist. In the following theorem, we introduce new variables that will allow us later to derive an approximate deflation method. The need to introduce new variables comes from the fact that in practice the exact root is not available, or even worse, the input coefficients contain small error. Therefore, our method shall seek for a slightly perturbed system with an exact multiple zero within a controlled neighborhood of the input, that fits as close as possible to the approximate multiplicity structure of the input system and point.

Theorem 5.6 (Deflation Theorem 2 [15]) *Let $\boldsymbol{f}(\boldsymbol{x})$ be a $n-$variate polynomial system with a $\mu-$fold isolated root at $\boldsymbol{x} = \boldsymbol{\zeta}$. The square system $\boldsymbol{\mathcal{D}}\tilde{\boldsymbol{g}}(\boldsymbol{x}, \tilde{\boldsymbol{\varepsilon}}) = 0$, as defined in (b), has a regular isolated root at $(\boldsymbol{x}, \tilde{\boldsymbol{\varepsilon}}) = (\boldsymbol{\zeta}, \mathbf{0})$.*

Nevertheless, this deflation does differ from the deflation strategy in [4, 12]. There, new variables are added that correspond to coefficients of differential elements, thus introducing a perturbation in the dual basis. This is suitable for exact equations, but, in case of perturbed data, the equations do not actually define a true singular point.

Example 5.10 Consider the system [13] of 3 equations in 2 variables $f_1 = x_1^3 + x_1x_2^2$, $f_2 = x_1x_2^2 + x_2^3$, $f_3 = x_1^2x_2 + x_1x_2^2$, and the singular point $(0, 0)$ of multiplicity equal to 7.

Suppose that the point is given. Using Theorem 5.3 and Lemma 5.2 we derive the primal-dual pair

$$\boldsymbol{\mathcal{D}} = (1, d_1, d_2, d_1^2, d_1d_2, d_2^2, \underline{d_2^3} + d_1^3 + d_1^2d_2 - d_1d_2^2)\ ,$$

where $\underline{d_2^3}$ is underlined to show that it corresponds to x_2^3 in the primal monomial basis $\boldsymbol{\mathcal{B}} = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2, x_2^3)$ The biggest matrix used, in depth 4, was of size 9×8, while Macaulay's method terminates with a matrix of size 30×15.

To deflate the root, we construct the augmented system $\boldsymbol{\mathcal{D}f}$ of 21 equations. The 21×2 Jacobian matrix $J_{\boldsymbol{\mathcal{D}f}}(\boldsymbol{x})$ is of rank 2 and a full-rank minor consists of the rows 4 and 5. Therefore, we find the system $(d_1^2[f_1], d_1d_2[f_1]) = (3x_1, 2x_2)$ which deflates $(0,0)$. Note that even though both equations of the deflated system derive from f_1, the functionals used on f_1 are computed using all initial equations.

The perturbed equations are then

$$\begin{aligned}
g_1 &= f_1 + \varepsilon_{1,1} + \varepsilon_{1,2}x_1 + \varepsilon_{1,3}x_2 + +\varepsilon_{1,4}x_2^2 + \varepsilon_{1,5}x_2^3 \\
g_2 &= f_2 + \varepsilon_{2,1} + \varepsilon_{2,2}x_1 + \varepsilon_{2,3}x_2 + +\varepsilon_{2,4}x_1^2 + \varepsilon_{2,5}x_1x_2 + \varepsilon_{2,6}x_2^2 + \varepsilon_{2,7}x_2^3 \\
g_3 &= f_3 + \uparrow_{3,1} + \uparrow_{3,2}x_1 + \uparrow_{3,3}x_2 + +\varepsilon_{3,4}x_1^2 + \varepsilon_{3,5}x_1x_2 + \varepsilon_{3,6}x_2^2 + \varepsilon_{3,7}x_2^3
\end{aligned}$$

and the resulting system $\boldsymbol{\mathcal{D}g}$ has a simple root at $(\boldsymbol{\zeta}, \mathbf{0})$.

5.5 Approximate Multiple Point

In real-life applications it is common to work with approximate inputs. Also, there is the need to (numerically) decide if an (approximate) system possesses a single (real) root in a given domain, notably for use in subdivision-based algorithms, e.g. [16,19].

In the regular case, Smale's $\alpha-$theory, extending Newton's method, can be used to answer this problem, also partially extended to singular cases in [7], using zero clustering. Another option is to used the following certification test, based on the verification method of Rump [22, Th. 2.1]:

Theorem 5.7 ([10,22] Krawczyk-Rump Theorem) *Let $\boldsymbol{f} \in R^n$, $R = \mathbb{K}[\boldsymbol{x}]$, be a polynomial system and $\boldsymbol{\zeta}^* \in \mathbb{R}^n$ a real approximate regular isolated point. Given an interval domain $Z \in \mathbb{IR}^n$ containing $\boldsymbol{\zeta}^* \in \mathbb{R}^n$, and an interval matrix $M \in \mathbb{IR}^{n\times n}$ whose $i-$th column M_i satisfies*

$$\nabla f_i(Z) \subseteq M_i \quad \textit{for} \quad i = 1\ldots,n$$

then the following holds: If the interval domain

$$V_{\boldsymbol{f}}(Z,\boldsymbol{\zeta}^*) := -J_{\boldsymbol{f}}(\boldsymbol{\zeta}^*)^{-1}\boldsymbol{f}(\boldsymbol{\zeta}^*) + (I - J_{\boldsymbol{f}}(\boldsymbol{\zeta}^*)^{-1}M)Z \tag{5.20}$$

is contained in $\overset{\circ}{Z}$, the interior of Z, then there is a unique $\boldsymbol{\zeta} \in Z$ with $\boldsymbol{f}(\boldsymbol{\zeta}) = \mathbf{0}$ and the Jacobian matrix $J_{\boldsymbol{f}}(\boldsymbol{\zeta}) \in M$ is non-singular.

In our implementation we use this latter approach, since it is suitable for inexact data and suits best with the perturbation which is applied. In particular, it coincides with the numerical scheme of [22] in the univariate case.

In the case of an isolated multiple point of a polynomial system, we applied a deflation to transform it into a regular root of an extended system. The theorem is applied to the system of Theorem 5.6, using an (approximate) structure $\boldsymbol{\mathcal{D}}$. The resulting range of the $\boldsymbol{\varepsilon}-$parameters encloses a system that attains a single multiple root of that structure. Hence the domain for $\boldsymbol{\varepsilon}-$variables reflects the distance of the input system from a precise system with local structure $\boldsymbol{\mathcal{D}}$. Therefore, we obtain a perturbed system in a neighborhood of the input together with a numerically controlled bound on the perturbation coefficients, with a unique multiple root having a prescribed multiplicity.

If the multiple point is known approximately, we use implicitly Taylor's expansion of the polynomials at this approximate point to deduce the dual basis, applying the algorithm of the previous section. The following computation can be applied:

- At each step, the solutions of linear system (5.10, i–iii) are computed via Singular Value Decomposition. Using a given threshold, we determine the numerical rank and an orthogonal basis of the solutions from the last singular values and the last columns of the right factor of the SVD.
- For the computation of the monomials which define the equations (Lemma 5.2, iii) at the next step, we apply QR decomposition on the transpose of the basis to extract a non-zero maximal minor. The monomials indexing this minor are used to determine constraints (5.10, i–iii). A similar numerical technique is employed in [25], for Macaulay's method.

Example 5.11 Let $f_1 = x_1^2x_2 - x_1x_2^2$, $f_2 = x_1 - x_2^2$. The verification method of [22] applies a linear perturbation to this system, but fails to certify the root $\boldsymbol{\zeta} = (0, 0)$.

We consider an approximate point $\boldsymbol{\zeta}^* = (.01, .002)$ and we compute the approximate multiplicity structure

$$\boldsymbol{\mathcal{D}} = (\Lambda_1, \dots, \Lambda_4) = (1.0, 1.0d_2, \underline{1.0d_1} + 1.0d_2^2, \underline{1.0d_1d_2} + 1.0d_2^3).$$

The augmented system $\boldsymbol{g}(\boldsymbol{x}) = (\Lambda_j[f_i]) = (f_1,\ 2.0x_1x_2 - 1.0x_2^2 - 1.0x_1,\ 2.0x_1 - 2.0x_2,\ 1.0x_1 - 1.0x_2^2,\ f_2,\ -2.0x_2,\ 0., 0.)$ has a Jacobian matrix:

$$J_{\boldsymbol{g}}(\boldsymbol{\zeta}^*)^T = \begin{bmatrix} .00 & .016 & -.99 & 2.0 & 1.0 & 0 & 0 & 0 \\ .00 & -.02 & .016 & -2.0 & -.004 & -2.0 & 0 & 0 \end{bmatrix}$$

with a non-zero minor at the third and forth row. Using this information, we apply the following perturbation to the original system:

$$g_1 = x_1^2x_2 - x_1x_2^2 + \varepsilon_{11} + \varepsilon_{12}x_2$$
$$g_5 = x_1 - x_2^2 + \varepsilon_{21} + \varepsilon_{22}x_2 + \varepsilon_{23}x_1 + \varepsilon_{24}x_1x_2$$

Thus $\boldsymbol{g}(x_1, x_2, \varepsilon_{11}, \varepsilon_{12}, \varepsilon_{21}, \varepsilon_{22}, \varepsilon_{23}, \varepsilon_{24})$, computed as before, is a square system with additional equations:

$$\begin{aligned}
g_2 &= 1.0x_1^2 - 2.0x_1x_2 + 1.0\varepsilon_{12}\\
g_3 &= 2.0x_1x_2 - 1.0x_2^2 - 1.0x_1\\
g_4 &= 2.0x_1 - 2.0x_2\\
g_6 &= -2.0x_2 + 1.0\varepsilon_{22} + 1.0x_1\varepsilon_{24}\\
g_7 &= 1.0\varepsilon_{23} + 1.0x_2\varepsilon_{24}\\
g_8 &= 1.0\varepsilon_{24}
\end{aligned}$$

Now take the box $Z_1 = [-.03, .05] \times [-.04, .04] \times [-.01, .01]^6$. We apply Theorem 5.7 on $\boldsymbol{g}$, i.e. we compute $V_{\boldsymbol{g}}(Z_1, \boldsymbol{\zeta}^*)$. For the variable ε_{21} the interval is $[-.015, .15] \not\subseteq (-.01, .01)$, therefore we don't get an answer.

We shrink a little Z_1 down to $Z_2 = [-.03, .05] \times [-.02, .02] \times [-.01, .01]^6$ and we apply again Theorem 5.7, which results in

$$V_{\boldsymbol{g}}(Z_2, (\boldsymbol{\zeta}^*, \mathbf{0})) = \begin{bmatrix} [-.004, .004] \\ [-.004, .004] \\ [-.001, .001] \\ [-.007, .007] \\ [-.006, .006] \\ [-.009, .009] \\ [-.00045, .00035] \\ [.0, .0] \end{bmatrix} \overset{\circ}{\subseteq} Z_2,$$

thus we certify that the input equations admit a perturbation of magnitude of .01, so that the perturbed system has a unique exact root within the interval $[-.03, .05] \times [-.02, .02]$.

5.6 Experimentation

We have implemented the presented algorithms in MAPLE. It can compute (approximate) dual bases by means of Macaulay's method as well as the integration method, and it can derive the augmented system defined in Theorem 5.6. Then Krawczyk-Rump's interval method is used to verify the root.

Example 5.12 Let, as in [11, 13],

$$\begin{aligned}
f_1 &= 2x_1 + 2x_1^2 + 2x_2 + 2x_2^2 + x_3^2 - 1,\\
f_2 &= (x_1 + x_2 - x_3 - 1)^3 - x_1^3,\\
f_3 &= (2x_1^3 + 2x_2^2 + 10x_3 + 5x_3^2 + 5)^3 - 1{,}000x_1^5.
\end{aligned}$$

The point $(0, 0, -1)$ occurs with multiplicity equal to 18, in depth 7. The final matrix size with our method is 54×37, while Macaulay's method ends with a 360×165 matrix.

If the objective is to deflate as efficiently as possible, then one can go step by step: First compute a basis of $\mathscr{D}_1$ and stop the process. We get the evaluation $\mathbf{1}$ and 2 first order functionals, which we apply to f_1. We arrive at

$$(\mathbf{1}[f_1], (d_2 - d_1)[f_1],\ (d_1 + d_3)[f_1]) = (f_1, -4x_1 + 4x_2, 2 + 4x_1 + 2x_3)$$

and we check that the Jacobian determinant is 64, thus we have a deflated system only with a partial local structure. The condition number of the Jacobian matrix is also very satisfactory, with a value of around 5.55.

The recent paper [8], implementing the dialytic deflation method produces a deflated system of size 75×48 for this instance, with a condition number of order 10^6.

Example 5.13 Consider the equations (taken from [4, DZ3]):

$$\begin{aligned}
f_1 &= 14x_1+33x_2-3\sqrt{5}(x_1^2+4x_1x_2+4x_2^2+2)+\sqrt{7}+x_1^3+6x_1^2x_2+12x_1x_2^2+8x_2^3,\\
f_2 &= 41x_1-18x_2-\sqrt{5}+8x_1^3-12x_1^2x_2+6x_1x_2^2-x_2^3+3\sqrt{7}(4x_1x_2-4x_1^2-x_2^2-2)
\end{aligned}$$

and take an approximate system $\tilde{\boldsymbol{f}}$ with those coefficients rounded to 6 digits. A 5−fold zero of $\boldsymbol{f}$ rounded to 6 digits is $\boldsymbol{\zeta}^* = (1.50551, .365278)$.

Starting with the approximate system and with a tolerance of .001, we compute the basis

$$\begin{aligned}
\boldsymbol{\mathcal{D}} = (&1, d_1 + .33d_2, d_1^2 + .33d_1d_2 + .11d_2^2, d_1^3 + .33d_1^2d_2 + .11d_1d_2^2 + .03d_2^3\\
&-1.54d_2, d_1^4+.33d_1^3d_2+.11d_1^2d_2^2+.03d_1d_2^3+.01d_2^4-1.54d_1d_2-1.03d_2^2)
\end{aligned}$$

having 4 correct digits, with respect to the initial exact system, and the primal counterpart $\boldsymbol{\mathcal{B}} = (1, x_1, x_1^2, x_1^3, x_1^4)$.

We form the deflated system (b), with $I = \{(3, 1), (5, 1)\}$, i.e. the 3rd and 5th dual element on f_1 have non-null Jacobian. By adding 8 new variables, the system is perturbed as:

$$\begin{aligned}
g_{1,1} &= \tilde{f}_1 + \varepsilon_{1,1} + \varepsilon_{1,2}(x_1 - \zeta_1^*) + \varepsilon_{1,4}(x_1 - \zeta_1^*)^3\ ,\\
g_{2,1} &= \tilde{f}_2 + \sum_{i=1}^{5} \varepsilon_{2,i}(x_1 - \zeta_1^*)^{i+1}
\end{aligned}$$

and their derivation with respect to $\boldsymbol{\mathcal{D}}$.

We consider a box Z with center $= \boldsymbol{\zeta}^*$ and length $= .004$ at each side. Also, we allow a range $E = [-.004, .004]^8$ for the variables $\tilde{\boldsymbol{\varepsilon}}$. Applying Theorem 5.7 we get

Table 5.1 Benchmark systems from [3], reporting matrix size at the last step of computing a dual local basis, and overall time for primal-dual computation. The computations are done using Maple. Observe that Macaulay's method results in a matrix of size $n\binom{p-1+n}{p-1}\times\binom{p+n}{p}$, in contrast to a matrix of size $(\frac{n(n-1)}{2}\boldsymbol{\mu}+n)\times\boldsymbol{\mu}(n-1)+1$ for the primal-dual approach

System	μ/n	**MM'11**		**Mourrain'97**		**Macaulay**	
cmbs1	11/3	27 × 23	.18 s	27 × 33	.95 s	105 × 56	1.55 s
cmbs2	8/3	21 × 17	.08 s	21 × 24	.39 s	60 × 35	.48 s
mth191	4/3	10 × 9	.03 s	10 × 12	.07 s	30 × 20	.14 s
decker2	4/2	5 × 5	.02 s	5 × 8	.05 s	20 × 15	.10 s
Ojika2	2/3	6 × 5	.02 s	6 × 6	.03 s	12 × 10	.04 s
Ojika3	4/3	12 × 9	.07 s	12 × 12	.27 s	60 × 35	.59 s
KSS	16/5	155 × 65	8.59 s	155 × 80	40.41 s	630 × 252	70.03 s
Capr.	4/4	22 × 13	.28 s	22 × 16	.47 s	60 × 35	2.34 s
Cyclic-9	4/9	104 × 33	1.04 s	104 × 36	5.47 s	495 × 220	31.40 s
DZ1	131/4	700 × 394	14 m	700 × 524	26 m	4,004 × 1,365	220 m
DZ2	16/3	43 × 33	.68 s	43 × 48	4.38 s	360 × 165	25.72 s
DZ3	5/2	6 × 6	.04 s	6 × 10	.23 s	30 × 21	.79 s

a verified inclusion $V_{\boldsymbol{g}}(Z\times E,(\boldsymbol{\zeta}^*,\mathbf{0}))$ inside $Z\times E$ and we deduce that a unique specialization $\tilde{\boldsymbol{\varepsilon}}\in E$ "fits" the approximate system $\tilde{\mathbf{f}}$ to the multiplicity structure $\boldsymbol{\mathcal{D}}$.

Indeed, one iteration of Newton's method on $\boldsymbol{g}(\boldsymbol{x},\boldsymbol{\varepsilon})$ gives the approximate point $\boldsymbol{\zeta}=(1.505535473, .365266196)$ and corresponding values for $\boldsymbol{\varepsilon}_0\in E$, such that $\boldsymbol{\zeta}$ is a 9–digit approximation of the multiple root of the perturbed system $\boldsymbol{g}(\boldsymbol{x},\boldsymbol{\varepsilon}_0)$.

In Table 5.1 we run dual basis computation on the benchmark set of [4]. Multiplicity, matrix sizes at termination step and computation time is reported. One sees that there is at least an order of gain in the running time using the primal-dual approach.

Acknowledgements This research has received funding from the EU's 7th Framework Programme [FP7/2007-2013], Marie Curie Initial Training Network SAGA, grant n° [PITN-GA-2008-214584].

References

1. L. Alberti, B. Mourrain, J. Wintz, Topology and arrangement computation of semi-algebraic planar curves. Comput. Aided Geom. Des. **25**, 631–651 (2008)
2. M.F. Atiyah, I.G. MacDonald, *Introduction to Commutative Algebra* (Addison-Wesley, Reading, 1969)
3. B.H. Dayton, T.-Y. Li, Z. Zeng, Multiple zeros of nonlinear systems. Math. Comput. **80**(276), 2143–2168 (2011)

4. B.H. Dayton, Z. Zeng, Computing the multiplicity structure in solving polynomial systems, in *ISSAC '05: Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation*, Beijing (ACM, New York, 2005), pp. 116–123
5. M. Elkadi, B. Mourrain, *Introduction à la résolution des systèmes d'équations algébriques*. Volume 59 of Mathématiques et Applications (Springer, Berlin/Heidelberg, 2007)
6. W.J. Gilbert, Newton's method for multiple roots. Comput. Graph. **18**(2), 227–229 (1994)
7. M. Giusti, G. Lecerf, B. Salvy, J.-C. Yakoubsohn, On location and approximation of clusters of zeros: case of embedding dimension one. Found. Comput. Math. **7**, 1–58 (2007). doi:10.1007/s10208-004-0159-5
8. W. Hao, A.J. Sommese, Z. Zeng, Algorithm 931: an algorithm and software for computing multiplicity structures at zeros of nonlinear systems. ACM Trans. Math. Softw. **40**(1):5:1–5:16 (2013)
9. L.V. Kantorovich, Functional analysis and applied mathematics. Uspekhi Matematicheskikh Nauk **3**(6), 89–185 (1948)
10. R. Krawczyk, Newton-algorithmen zur bestimmung von nullstellen mit fehlerschranken. Computing **4**(3), 187–201 (1969)
11. G. Lecerf, Quadratic newton iteration for systems with multiplicity. Found. Comput. Math. **2**, 247–293 (2002)
12. A. Leykin, J. Verschelde, A. Zhao, Newton's method with deflation for isolated singularities of polynomial systems. Theor. Comput. Sci. **359**(1–3), 111–122 (2006)
13. A. Leykin, J. Verschelde, A. Zhao, Higher-order deflation for polynomial systems with isolated singular solutions, in *Algorithms in Algebraic Geometry*, ed. by A. Dickenstein, F.-O. Schreyer, A.J. Sommese. Volume 146 of The IMA Volumes in Mathematics and Its Applications (Springer, New York, 2008), pp. 79–97
14. F.S. Macaulay, *The Algebraic Theory of Modular Systems* (Cambridge University Press, Cambridge, 1916)
15. A. Mantzaflaris, B. Mourrain, Deflation and certified isolation of singular zeros of polynomial systems, in *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, San Jose, June 2011, ed. by A. Leykin (ACM, New York, 2011), pp. 249–256
16. A. Mantzaflaris, B. Mourrain, E. Tsigaridas, Continued fraction expansion of real roots of polynomial systems, in *Proceedings of the 2009 Conference on Symbolic-Numeric Computation, SNC '09*, Kyoto (ACM, New York, 2009), pp. 85–94
17. M.G. Marinari, T. Mora, H.M. Möller, Gröbner duality and multiplicities in polynomial system solving, in *Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation, ISSAC '95*, Montreal (ACM, New York, 1995), pp. 167–179
18. B. Mourrain, Isolated points, duality and residues. J. Pure Appl. Algebra **117–118**, 469–493 (1997)
19. B. Mourrain, J.P. Pavone, Subdivision methods for solving polynomial equations. J. Symb. Comput. **44**, 292–306 (2009)
20. T. Ojika, S. Watanabe, T. Mitsui, Deflation algorithm for the multiple roots of a system of nonlinear equations. J. Math. Anal. Appl. **96**(2), 463–479 (1983)
21. S.R. Pope, A. Szanto, Nearest multivariate system with given root multiplicities. J. Symb. Comput. **44**(6), 606–625 (2009)
22. S. Rump, S. Graillat, Verified error bounds for multiple roots of systems of nonlinear equations. Numer. Algorithms **54**, 359–377 (2010). doi:10.1007/s11075-009-9339-3
23. H.J. Stetter, Analysis of zero clusters in multivariate polynomial systems, in *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation, ISSAC '96*, Zurich (ACM, New York, 1996), pp. 127–136
24. X. Wu, L. Zhi, Computing the multiplicity structure from geometric involutive form, in *Proceedings of the Twenty-First International Symposium on Symbolic and Algebraic Computation, ISSAC '08*, Hagenberg (ACM, New York, 2008), pp. 325–332
25. Z. Zeng, The closedness subspace method for computing the multiplicity structure of a polynomial system, in *Interactions of Classical and Numerical Algebraic Geometry*, ed. by D. Bates, G. Besana, S. Di Rocco, C. Wampler. Volume 496 of Contemporaty Mathematics (American Mathematical Society, Providence, 2009), pp. 347–362

Chapter 6
Plane Mixed Discriminants and Toric Jacobians

Alicia Dickenstein, Ioannis Z. Emiris, and Anna Karasoulou

Dedicated to the memory of our friend Andrei Zelevinsky (1953–2013)

6.1 Introduction

Polynomial algebra offers a standard and powerful approach to handle several problems in geometric modeling. In particular, the study and solution of systems of polynomial equations has been a major topic. Discriminants provide a key tool when examining well-constrained systems, including the case of one univariate polynomial. Their theoretical study is a thriving and fruitful domain today, but they are also very useful in a variety of applications.

The best studied discriminant is probably known since high school, where one studies the discriminant of a quadratic polynomial $f(x) = ax^2+bx+c = 0\,(a \neq 0)$. The polynomial f has a double root if and only if its discriminant $\Delta_2 = b^2 - 4ac$ is equal to zero. Equivalently, this can be defined as the condition for $f(x)$ and its derivative $f'(x)$ to have a common root:

$$\exists\, x \;:\; f(x) = ax^2 + bx + c \,=\, f'(x) = 2ax + b = 0 \;\Leftrightarrow\; \Delta_2 = 0. \tag{6.1}$$

One can similarly consider the discriminant of a univariate polynomial of any degree. If we wish to calculate the discriminant $\Delta_5(f)$ of a polynomial f of degree five in one variable, we consider the condition that both f and its derivative vanish:

$$\begin{aligned} f(x) &= ax^5 + bx^4 + cx^3 + dx^2 + ex + g = 0,\\ f'(x) &= 5ax^4 + 4bx^3 + 3cx^2 + 2dx + e = 0. \end{aligned}$$

A. Dickenstein (✉)
Department of Mathematics, FCEN, IMAS, CONICET, University of Buenos Aires, Buenos Aires, Argentina
e-mail: alidick@dm.uba.ar

I.Z. Emiris • A. Karasoulou
Department of Informatics and Telecommunications, University of Athens, Athens, Greece
e-mail: emiris@di.uoa.gr; akarasou@di.uoa.gr

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_6

In this case, elimination theory reduces the computation of Δ_5 to the computation of a 9×9 Sylvester determinant, which equals $a\,\Delta_5(f)$. If we develop this determinant, we find out that the number monomials in the discriminant increases rapidly with the input degree:

$$\begin{aligned}
&\Delta_5 = -2{,}050a^2g^2bedc + 356abed^2c^2g - 80b^3ed^2cg + 18dc^3b^2g \\
&e - 746agdcb^2e^2 + 144ab^2e^4c - 6ab^2e^3d^2 - 192a^2be^4d - 4d^2ac \\
&{}^3e^2 + 144d^2a^2ce^3 - 4d^3b^3e^2 - 4c^3e^3b^2 - 80abe^3dc^2 + 18b^3e^3 \\
&dc + 18d^3acbe^2 + d^2c^2b^2e^2 - 27b^4e^4 - 128a^2e^4c^2 + 16ac^4e^3 - 27 \\
&a^2d^4e^2 + 256a^3e^5 + 3{,}125a^4g^4 + 160a^2gbe^3c + 560a^2gdc^2e^2 + 1{,}020 \\
&a^2gbd^2e^2 + 160ag^2b^3ed + 560ag^2d^2cb^2 + 1{,}020ag^2b^2c^2e - 192 \\
&b^4ecg^2 + 24ab^2ed^3g + 24abe^2c^3g + 144b^4e^2dg - 6b^3e^2c^2g + 14 \\
&4dc^2b^3g^2 - 630dac^3bg^2 - 630d^3a^2ceg - 72d^4acbg - 72dac^4e \\
&g - 4d^3c^2b^2g - 1{,}600ag^3cb^3 - 2{,}500a^3g^3be - 50a^2g^2b^2e^2 - 3{,}750a^3 \\
&g^3dc + 2{,}000a^2g^3db^2 + 2{,}000a^3g^2ce^2 + 825a^2g^2d^2c^2 + 2{,}250a^2g^3b \\
&c^2 + 2{,}250a^3g^2ed^2 - 900a^2g^2bd^3 - 900a^2g^2c^3e - 36agb^3e^3 - 1{,}600 \\
&a^3ge^3d + 16d^3ac^3g - 128d^2b^4g^2 + 16d^4b^3g - 27c^4b^2g^2 + 108ac^5 \\
&g^2 + 108a^2d^5g + 256b^5g^3.
\end{aligned}$$

In fact, if we compute the resultant of f and xf' by means of the 10×10 Sylvester determinant, we find the more symmetric output: $a\,g\,\Delta_5(f)$. This formula is very well known for univariate discriminants (Ch.12, [18]), and we generalize it in Theorem 3.

One univariate polynomial is the smallest well-constrained system. We are concerned with multivariate systems of sparse polynomials, in other words, polynomials with fixed support, or set of nonzero terms. *Sparse (or toric)* elimination theory concerns the study of resultants and discriminants associated with toric varieties. This theory has its origin in the work of Gel'fand, Kapranov and Zelevinsky on multivariate hypergeometric functions. Discriminants arise as singularities of such functions [17].

Gel'fand, Kapranov and Zelevinsky [18] established a general definition of sparse discriminant, which gives as special case the following definition of (sparse) mixed discriminant (see Sect. 6.2 for the relation with the discriminant of the associated Cayley matrix and with the notion of mixed discriminant in [3]). In case $n = 2$, the mixed discriminant detects tangencies between families of curves with fixed supports. In general, the *mixed discriminant* $\Delta_{A_1,\dots,A_n}(f_1,\dots,f_n)$ of n polynomials in n variables with fixed supports $A_1,\dots,A_n \subset \mathbb{Z}^n$ is the irreducible polynomial (with integer coprime coefficients, defined up to sign) in the coefficients of the f_i which vanishes whenever the system $f_1 = \cdots = f_n = 0$ has a multiple root (that is, a root which is not simple) with non-zero coordinates, in case this discriminantal variety is a hypersurface (and equal to the constant 1 otherwise). The zero locus of the mixed discriminant is the variety of ill-posed systems [24].

For each $i = 1,\dots,n$, pick an element $a_{i,0} \in A_i$ and denote by $\mathcal{L}_{A_1,\dots,A_n}$ the lattice generated by $\{a - a_{i,0}, a \in A_i, i = 1,\dots,n\}$. We shall work with the polynomial defining the *discriminant cycle* (see Sect. 6.2), which is defined as the power $\Delta_{A_1,\dots,A_n}^{i(A_1,\dots,A_n)}$ of the mixed discriminant raised to the index

$$i(A_1,\dots,A_n) = [\mathbb{Z}^n : \mathcal{L}_{A_1,\dots,A_n}], \tag{6.2}$$

which we always assume to be finite. In most situations, this index equals 1 and so both concepts coincide.

Discriminants have many applications. Besides the classical application in the realm of differential equations to describe singularities, discriminants occur for instance in the description of the topology of real algebraic plane curves [19], in solving systems of polynomial inequalities and zero-dimensional systems [16], in determining the number of real roots of square systems of sparse polynomials [9], in studying the stability of numerical solving [6], in the computation of the Voronoi diagram of curved objects [13], or in the determination of cusp points of parallel manipulators [20].

Computing (mixed) discriminants is a (difficult) elimination problem. In principle, they can be computed with Gröbner bases, but this is very inefficient in general since these polynomials have a rich combinatorial structure [18]. Ad-hoc computations via complexes (i.e., via tailored homological algebra) are also possible, but they also turn out to be complicated. The tropical approach to compute discriminants was initiated in [8] and the tropicalization of mixed planar discriminants was described in [10]. Recently, in [12], the authors focus on computing the discriminant of a multivariate polynomial via interpolation, based on [11, 23]; the latter essentially offers an algorithm for predicting the discriminant's Newton polytope, hence its nonzero terms. This yields a new output-sensitive algorithm which, however, remains to be juxtaposed in practice to earlier approaches.

We mainly work in the case $n = 2$, where the results are more transparent and the basic ideas are already present, but all our results and methods can be generalized to any number of variables. This will be addressed in a subsequent paper [7]. Consider for instance a system of two polynomials in two variables and assume that, the first polynomial factors as $f_1 = f_1' \cdot f_1''$. Then, the discriminant also factors and we thus obtain a multiplicativity formula for it, which we make precise in Corollary 7. This significantly simplifies the discriminant's computation and generalizes the formula in [2] for the classical homogeneous case. This multiplicativity formula is a consequence of our main result (Theorem 3 in dimension 2, see also Theorem 4 in any dimension) relating the mixed discriminant and the resultant of the given polynomials and their *toric Jacobian* (see Sect. 6.3 for precise definitions and statements). As another consequence of Theorem 3, we reprove, in Corollary 6, the bidegree formula for planar mixed discriminants in [3].

The rest of this chapter is organized as follows. The next section overviews relevant existing work and definitions. In Sect. 6.3 we present our main results relating the mixed discriminant with the sparse resultant of the two polynomials and their toric Jacobian. In Sect. 6.4 we deduce the general multiplicativity formula for the mixed discriminant when one polynomial factors.

6.2 Previous Work and Notation

In this section we give a general description of discriminants and some definitions and notations that we are going to use in the following sections.

Given a set $A \subset \mathbb{R}^n$, let $Q = conv(A)$ denote the convex hull of A. We say that A is a *lattice configuration* if it is contained in $\mathbb{Z}^n$, whereas a polytope with integer vertices is called a lattice polytope. We denote by $\mathrm{Vol}(\cdot)$ the volume of a lattice polytope, normalized with respect to the lattice $\mathbb{Z}^n$, so that a primitive simplex has normalized volume equal to 1. Normalized volume is obtained by multiplying Euclidean volume by $n!$.

Given a non-zero Laurent polynomial

$$f = \sum_a c_a x^a,$$

the finite subset A of $\mathbb{Z}^n$ of those exponents a for which $c_a \neq 0$ is called the *support* of f. The *Newton polytope* $N(f)$ of f is the lattice polytope defined as the convex hull of A. We will assume that the coefficients c_a take values in an algebraically closed field K of characteristic 0.

A (finite) set A is said to be *full*, if it consists of all the lattice points in its convex hull. In [3], A is called *dense* in this case, but we prefer to reserve the word dense to refer to the classical homogeneous case. A subset $F \subseteq A$ is called a *face* of A, denoted $F \prec A$, if F is the intersection of A with a face of the polytope $conv(A)$.

As usual $Q_1 + Q_2$ denotes the Minkowski sum of sets Q_1 and Q_2 in $\mathbb{R}^n$. The *mixed volume* $MV(Q_1, \dots, Q_n)$ of n convex polytopes Q_i in $\mathbb{R}^n$ is the multilinear symmetric function with respect to Minkowski sum that generalizes the notion of volume in the sense that $MV(Q, \dots, Q) = \mathrm{Vol}(Q)$, when all Q_i are equal to a fixed convex polytope Q.

The following key result is due to Bernstein and Kouchnirenko. The mixed volume of the Newton polytopes of n Laurent polynomials $f_1(x), \dots, f_n(x)$ in n variables is an integer that bounds the number of isolated common solutions of $f_1(x) = 0, \dots, f_n(x) = 0$ in the algebraic torus $(K^*)^n$, over an algebraically closed field K of characteristic 0 containing the coefficients. If the coefficients of the polynomials are generic, then the common solutions are isolated and their number equals the mixed volume. This bound generalizes Bézout's classical bound to the sparse case: for homogeneous polynomials the mixed volume and Bézout's bound coincide.

Mixed volume can be defined in terms of Minkowski sum volumes as follows.

$$MV(Q_1, \dots, Q_n) = \sum_{k=1}^{n} (-1)^{n-k} \sum_{I \subset \{1,\dots,n\}, |I|=k} \frac{1}{n!} \mathrm{Vol}\Big(\sum_{i \in I} Q_i\Big).$$

This implies, for $n = 2$:

$$2MV(Q_1, Q_2) = \mathrm{Vol}(Q_1 + Q_2) - \mathrm{Vol}(Q_1) - \mathrm{Vol}(Q_2).$$

Definition 1 *A family of finite lattice configurations $A_1, \dots, A_k$ in $\mathbb{Z}^n$ is called essential if the affine dimension of the lattice $\mathcal{L}_{A_1,\dots,A_n}$ equals $k-1$, and for all proper subsets $I \subset \{1, \dots, k\}$ it holds that the dimension of the lattice generated by $\{a - a_i, a \in A_i, i \in I\}$ is greater or equal than its cardinality $|I|$.*

Definition/Theorem 1 [18, 25] *Fix a family of $n+1$ finite lattice configurations $A_1, \dots, A_{n+1}$ which contains a unique essential subfamily $\{A_i, i \in I\}$. Given Laurent polynomials $f_1, \dots, f_{n+1}$ in n variables with respective supports $A_1, \dots, A_{n+1}$, the resultant $\mathrm{Res}_{A_1,\dots,A_{n+1}}(f_1, \dots, f_{n+1})$ is the irreducible polynomial with coprime integer coefficients (defined up to sign) in the coefficients of $f_1, \dots, f_{n+1}$, that vanishes whenever $f_1, \dots, f_{n+1}$ have a common root in the torus $(\mathbb{C}^*)^n$. In fact, in this case, the resultant only depends on the coefficients of f_i with $i \in I$.*

If there exist two different essential subfamilies, then the (closure of the) variety of solvable systems is not a hypersurface and in this case we set:

$$\mathrm{Res}_{A_1,\dots,A_{n+1}}(f_1, \dots, f_{n+1}) = 1.$$

In what follows, we consider n (finite) lattice configurations $A_1, \dots, A_n$ in $\mathbb{Z}^n$ and we denote by $Q_1, \dots, Q_n$ their respective convex hulls. Let $f_1, \dots, f_n$ be Laurent polynomials with coefficients in K and support $A_1, \dots, A_n$, respectively:

$$f_i(x) = \sum_{\alpha \in A_i} c_{i,\alpha} x^\alpha, \quad i = 1 \dots, n.$$

In [3] the *mixed discriminantal variety*, is defined as closure of the locus of coefficients $c_{i,\alpha}$ for which the associated system $f_1 = \cdots = f_n = 0$ has a non-degenerate multiple root $x \in (K^*)^n$. This means that x is an isolated root and the n gradient vectors

$$\left(\frac{\partial f_i}{\partial x_1}(x), \dots, \frac{\partial f_i}{\partial x_n}(x)\right)$$

are linearly dependent, but any $n-1$ of them are linearly independent.

Definition 2 *If the mixed discriminantal variety is a hypersurface, the mixed discriminant of the previous system is the unique up to sign irreducible polynomial $\Delta_{A_1,\dots,A_n}$ with integer coefficients in the unknowns $c_{i,a}$ which defines this hypersurface. Otherwise, the family is said to be defective and we set $\Delta_{A_1,\dots,A_n} = 1$. The* mixed discriminant cycle *$\tilde{\Delta}_{A_1,\dots,A_n}$ is equal to $i(A_1, \dots, A_n)$ times the mixed discriminant variety, and thus its equation equals $\Delta_{A_1,\dots,A_n}$ raised to this integer (defined by (6.2)).*

By [3, Theorem 2.1], when the family $A_1, \dots, A_n$ is non defective, the mixed discriminant $\Delta_{A_1,\dots,A_n}$ coincides with the A-discriminant defined in [18], where A is the Cayley matrix

$$A = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \\ A_1 & A_2 & \dots & A_n \end{pmatrix}.$$

This matrix has $2n$ rows and $m = \sum_{i=1}^{n} |A_i|$ columns, so $0 = (0, \dots, 0)$ and $1 = (1, \dots, 1)$ denote row vectors of appropriate lengths. We introduce n new variables $y_1, \dots, y_n$ in order to encode the system $f_1 = \dots = f_n = 0$ in one polynomial with support in A, via the *Cayley trick*: $\phi(x, y) = y_1 f_1(x) + \dots + y_n f_n(x)$. Note that $i(A_1, \dots, A_n)$ equals the index in $\mathbb{Z}^{2n}$ of the lattice generated by the columns of the matrix A.

In what follows, when we refer to resultants or discriminants, we shall refer to the *equations of the corresponding cycles* (as in Definition 2), but we will omit the tildes in our notation. More explicitly, we will follow the convention in the articles [5] by D'Andrea and Sombra and [14] by Esterov, which is faithful to intersection theory. This convention allows us to present cleaner formulas. For instance, when the family $A_1, \dots, A_{n+1}$ is essential, our notion of resultant equals the resultant in [18, 25] raised to the index $i(A_1, \dots, A_{n+1})$. In most examples these two lattices coincide, and so our resultant cycle equals the resultant variety and the associated resultant polynomial is irreducible.

Remark 6.2 Assume A_1 consists of a single point α and that $\{1\}$ is the only essential subfamily of a given family $A_1, \dots, A_{n+1}$. Let $f_1(x) = cx^{\alpha}$. Then, for any choice of Laurent polynomials $f_2, \dots, f_{n+1}$ with supports $A_2, \dots, A_{n+1}$, it holds that (cf. [5, Example 3.14])

$$\mathrm{Res}_{A_1,\dots,A_{n+1}}(f_1, \dots, f_n) = c^{MV(A_2,\dots,A_{n+1})}. \tag{6.3}$$

With this convention, the following multiplicativity formula holds:

Theorem 2 [5, Corollary 4.6],[22] *Let $A_1', A_1'', A_1, \dots, A_{n+1}$ be finite subsets of $\mathbb{Z}^n$ with $A_1 = A_1' + A_1''$. Let $f_1, \dots, f_{n+1}$ be polynomials with supports contained in $A_1, \dots, A_{n+1}$ and assume that $f_1 = f_1' f_1''$ where f_1' has support A_1' and f_1'' has support A_1''. Then*

$$\begin{aligned} &\mathrm{Res}_{A_1,\dots,A_{n+1}}(f_1, \dots, f_{n+1}) \\ &\quad = \mathrm{Res}_{A_1',\dots,A_{n+1}}(f_1', \dots, f_{n+1}) \cdot \mathrm{Res}_{A_1'',\dots,A_{n+1}}(f_1'', \dots, f_{n+1}). \end{aligned}$$

Cattani, Cueto, Dickenstein, Di Rocco and Sturmfels in [3] proved that the degree of the mixed discriminant Δ is a piecewise linear function in the Plücker coordinates of a mixed Grassmanian. An explicit degree formula for plane curves is also presented in [3, Corollary 3.15]. In case A_1, A_2, they are two dimensional, full, and with the same normal fan, then the bidegree of Δ_{A_1,A_2} in the coefficients of f_1 and f_2 equals:

$$(\mathrm{Vol}(Q_1+Q_2)-area(Q_1)-perim(Q_2), \mathrm{Vol}(Q_1+Q_2)-area(Q_2)-perim(Q_1),$$

where $Q_i = conv(A_i)$, $i = 1, 2$, and $Q_1 + Q_2$ is their Minkowski sum. The area is normalized, so that a primitive triangle has area 1 and the perimeter $perim(Q_i)$ of Q_i is the cardinality of $\partial Q_i \cap \mathbb{Z}^2$. We will recover the general formula for this degree and present it in Corollary 6.

Busé and Jouanolou consider in [2] the following equivalent definition of the mixed discriminant, in case that $f_1, \dots, f_n$ are dense homogeneous polynomials in $(x_0, \dots, x_n)$ of degrees $d_1, \dots, d_n$ respectively, that is, their respective supports $A_i = d_i\sigma$ are all the lattice points in the d_i-th dilate of the unit simplex σ in $\mathbb{R}^n$. It is the non-zero polynomial in the coefficients of $f_1, \dots, f_n$ which equals

$$\frac{\mathrm{Res}_{d_1\sigma,\dots,d_n\sigma,\delta_i\sigma}(f_1, \dots, f_n, J_i)}{\mathrm{Res}_{d_1\sigma,\dots,d_n\sigma,\sigma}(f_1, \dots, f_n, x_i)}, \tag{6.4}$$

for all $i \in \{1, \dots, n\}$, where J_i is the maximal minor of the Jacobian matrix associated to $f_1, \dots, f_n$ obtained by deleting the i-th row and column and $\delta_i = \sum_{j\neq i}(d_j - 1)$. We give a more symmetric and general formula in Corollary 5 below.

The multiplicativity property of the discriminant in the case of dense homogeneous polynomials was already known to Sylvester in the multivariate case [26] and generalized by Busé and Jouanolou in [2], where they develop a formalism for discriminants for polynomials with coefficients in a ring. In particular $A_1 = d_1\sigma = (d_1' + d_1'')\sigma$ and f_1 is equal to the product $f_1' \cdot f_1''$ of two polynomials with respective degrees d_1', d_1'', the following factorization holds:

$$\begin{aligned}\Delta_{d_1\sigma,\dots,d_n\sigma}(f_1, \dots, f_n) &= \Delta_{d_1'\sigma,\dots,d_n\sigma}(f_1', \dots, f_n) \cdot \Delta_{d_1''\sigma,\dots,d_n\sigma}(f_1'', \dots, f_n) \\ &\quad \cdot \mathrm{Res}_{d_1'\sigma,d_1''\sigma,\dots,d_n\sigma}(f_1', f_1'', \dots, f_n)^2.\end{aligned} \tag{6.5}$$

It is straightforward to see from the definition, that in case $\Delta_{A_1',\dots,A_n}(f_1', \dots, f_n) = 0$ or $\Delta_{A_1'',\dots,A_n}(f_1'', \dots, f_n) = 0$ or $\mathrm{Res}_{A_1',A_1'',\dots,A_n}(f_1', f_1'', \dots, f_n) = 0$ then,

$$\Delta_{A_1'+A_1'',\dots,A_n}(f_1'f_1'', f_2, \dots, f_n) = 0.$$

It follows from [14] that when each support configuration A_i is full, the Newton polytope of the discriminant $\Delta_{A_1'+A_1'',A_2,\dots,A_n}(f_1'f_1'', f_2, \dots, f_n)$ equals the Minkowski sum of the Newton polytopes of the discriminants $\Delta_{A_1',A_2,\dots,A_n}(f_1', f_2, \dots, f_n)$ and $\Delta_{A_1'',A_2,\dots,A_n}(f_1'', f_2, \dots, f_n)$ plus two times the

Newton polytope of the resultant $\mathrm{Res}_{A_1', A_1'', A_2, \ldots, A_n}(f_1', f_1'', f_2, \ldots, f_n)$. So, a first guess would be that the factorization into the three factors in (6.5) above holds for general supports. We will see in Corollary 7 that other factors may occur, which we describe explicitly.

This behaviour already occurs in the univariate case:

Example 1 *Let $A_1' = \{0, i_1, \ldots, i_m, d_1\}$, $A_1'' = \{0, j_1, \ldots, j_l, d_2\}$ be the support sets of $f_1' = a_0 + a_{i_1}x^{i_1} + \cdots + a_{i_m}x^{i_m} + a_{d_1}x^{d_1}$, $f_1'' = b_0 + b_{j_1}x^{j_1} + \cdots + b_{j_l}x^{j_l} + b_{d_2}x^{d_2}$ respectively. Then*

$$\varDelta(f_1' f_1'') = \varDelta(f_1') \cdot \varDelta(f_1'') \cdot R(f_1', f_1'')^2 \cdot E,$$

where $E = a_0^{i_1 - m_0}\, b_0^{j_1 - m_0}\, a_{d_1}^{d_1 - i_m - m_1}\, b_{d_2}^{d_2 - j_l - m_1}$, with $m_0 := \min\{i_1, j_1\}$ and $m_1 := \min\{d_1 - i_m, d_2 - j_l\}$. On the other hand, in the full case $i_1 = j_1 = 1, i_m = d_1 - 1, j_l = d_2 - 1$, thus $E = 1$ because its exponents are equal to zero.

6.3 A General Formula

The aim of this section is to present a formula which relates the mixed discriminant with the resultant of the given polynomials and their toric Jacobian, whose definition we recall.

Definition 3 *Let $f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n)$ be n Laurent polynomials in n variables. The associated toric Jacobian J_f^T equals $x_1 \cdots x_n$ times the determinant of the Jacobian matrix of f, or equivalently, the determinant of the matrix:*

$$\begin{bmatrix} x_1 \dfrac{\partial f_1}{\partial x_1} & \cdots & x_n \dfrac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ x_1 \dfrac{\partial f_n}{\partial x_1} & \cdots & x_n \dfrac{\partial f_n}{\partial x_n} \end{bmatrix}.$$

Note that the Newton polytope of J_f^T is contained in the sum of the Newton polytopes of $f_1, \ldots, f_n$.

As we remarked before, we will mainly deal in this chapter with the case $n = 2$. Also, to avoid excessive notation and make the main results cleaner, we assume below that A_1, A_2 are two finite lattice configurations whose convex hulls satisfy

$$\dim(Q_1) = \dim(Q_2) = 2.$$

Let f_1, f_2 be polynomials with respective supports A_1, A_2:

$$f_i(x) = \sum_{\alpha \in A_i} c_{i,\alpha} x^{\alpha}, \quad i = 1, 2,$$

where $x = (x_1, x_2)$. We denote by Σ the set of primitive inner normals $\eta \in (\mathbb{Z}^2)^*$ of the edges of the convex hull of $A_1 + A_2$. We call A_i^η the face of A_i where the inner product with η is minimized. We call this minimum value ν_i^η. We also denote by f_i^η the subsum of terms in f_i with exponents in this face

$$f_i^\eta(x) = \sum_{\alpha \in A_i^\eta} c_{i,\alpha} x^\alpha, \quad i = 1, 2,$$

which is η-homogeneous of degree ν_i^η. Now, A_i^η is either a vertex of A_i (but not of both A_1, A_2 since two vertices do not give a Minkowski sum edge), or its convex hull is an edge of A_i (with inner normal η), which we denote by e_i^η. Note that if the face of $A_1 + A_2$ associated to η is a vertex, both polynomials f_i^η are monomials and their resultant locus has codimension two.

We denote by $\mu_i(\eta)$ $(i = 1, 2)$ the integer defined by the following difference:

$$\mu_i(\eta) = \min\{\langle \eta, m \rangle, m \in A_i - A_i^\eta\} - \nu_i^\eta \tag{6.6}$$

and by

$$\mu(\eta) = \min\{\mu_1(\eta), \mu_2(\eta)\}, \tag{6.7}$$

the minimum of these two integers. Note that by our assumption that $\dim(Q_i) = 2$, we have that $\mu(\eta) \geq 1$.

Without loss of generality, we can translate the support sets A_1^η, A_2^η to the origin and consider the line L^η containing them. The resultant (cycle) $\mathrm{Res}_{A_1^\eta, A_2^\eta}(f_1^\eta, f_2^\eta)$ is considered as before, with respect to the lattice $L^\eta \cap \mathbb{Z}^2$.

Remark 6.3 As in Remark 6.2, if f_1^η is a monomial, the resultant equals the coefficient of f_1^η raised to the normalized length $\ell(e_2^\eta)$ of the edge e_2^η of A_2 (that is, the number of integer points in the edge, minus 1). If η is an inner normal of edges A_1^η and A_2^η, pick points $a_{i,0}^\eta \in A_i^\eta, i = 1, 2$, the resultant we consider equals the irreducible resultant raised to the index of the lattice generated by $\{a - a_{i,0}^\eta, a \in A_i^\eta, i = 1, 2\}$ in $L^\eta \cap \mathbb{Z}^2$. Note that the exponent $\mu(\eta) = 1$ if at least one of the configurations is full.

The following is our main result. We present a rather complete sketch of the proof; a full proof requires further technical tools related to the notions in [15], which will be given for the general case in [7]. We recall our convention that resultants and discriminants are defined as the irreducible equations raised to the lattice indices that define the corresponding cycles.

Theorem 3 *Let f_1, f_2 be generic Laurent polynomials with respective supports A_1, A_2. Then, we have the following equality of polynomials up to a nonzero rational number:*

$$\mathrm{Res}_{A_1,A_2,A_1+A_2}(f_1, f_2, J_f^T) = \Delta_{A_1,A_2}(f_1, f_2) \cdot E,$$

where the factor E equals the finite product:

$$E = \prod_{\eta \in \Sigma} \mathrm{Res}_{A_1^\eta, A_2^\eta}(f_1^\eta, f_2^\eta)^{\mu(\eta)}.$$

Proof. Let X be the projective toric variety associated to $A_1 + A_2$. This compact variety consists of an open dense set T_X isomorphic to the torus $(\mathbb{C}^*)^2$ plus one toric Weil divisor D_η for each $\eta \in \Sigma$. The Laurent polynomials f_1, f_2, J_f^T define sections L_1, L_2, L_J of globally generated line bundles on X. The resultant $\mathrm{Res}_{A_1,A_2,A_1+A_2}(f_1, f_2, J_f^T)$ vanishes if and only if L_1, L_2, L_J have a common zero on X, which could be at T_X or at any of the D_η. This indicates the only possible factors of the resultant.

There is an intersection point at T_X if and only if there is a common zero of f_1, f_2 and J_f^T in the torus $(\mathbb{C}^*)^2$. In this case, the discriminant $\Delta_{A_1,A_2}(f_1, f_2)$ vanishes. It follows that $\Delta_{A_1,A_2}(f_1, f_2)$ divides $\mathrm{Res}_{A_1,A_2,A_1+A_2}(f_1, f_2, J_f^T)$ (the indices $[\mathbb{Z}^2 : \mathcal{L}_{A_1,A_2}]$ and $[\mathbb{Z}^2 : \mathcal{L}_{A_1,A_2,A_1+A_2}]$ are equal).

If instead there is a common zero at some D_η, this translates into the fact that f_1^η, f_2^η and $(J_f^T)^\eta = J_{f^\eta}^T$ (with obvious definition) have a common solution. But as f_i^η are η-homogeneous, they satisfy the weighted Euler equalities:

$$\eta_1 x_1 \frac{\partial f_i^\eta}{\partial x_1} + \eta_2 x_2 \frac{\partial f_i^\eta}{\partial x_2} = \nu_i^\eta f_i^\eta, \quad i = 1, 2, \tag{6.8}$$

from which we deduce that $J_{f^\eta}^T$ lies in the ideal $I(f_1^\eta, f_2^\eta)$ and so, the three polynomials will vanish exactly when there is a nontrivial common zero of f_1^η and f_2^η. This implies that all facet resultants $\mathrm{Res}_{A_1^\eta, A_2^\eta}(f_1^\eta, f_2^\eta)$ divide $\mathrm{Res}_{A_1,A_2,A_1+A_2}(f_1, f_2, J_f^T)$.

Now, we wish to see that the resultant $\mathrm{Res}_{A_1^\eta, A_2^\eta}(f_1^\eta, f_2^\eta)$ raised to the power $\mu(\eta)$ occurs as a factor. The following argument would be better written in terms of the multihomogeneous polynomials in the Cox coordinates of X which represent L_1, L_2, L_J [4]. Fix a primitive inner normal direction $\eta \in \Sigma$ of $A_1 + A_2$, let t be a new variable and define the following polynomials

$$F_i(t, x) = \sum_{\alpha \in A_i} c_{i,\alpha} t^{\langle \eta, \alpha \rangle - \nu_i^\eta} x^\alpha, \quad i = 1, 2, \tag{6.9}$$

so that

$$F_i(1, x) = f_i(x), \quad F_i(0, x) = f_i^\eta(x), \quad i = 1, 2,$$

and we can write

$$F_i(t,x) = f_i^{\eta}(t,x) + t^{\mu_i(\eta)} g_i(x) + t^{\mu_i(\eta)+1} h_i(t,x), \quad i = 1,2, \tag{6.10}$$

where the polynomials $g_i(x)$ and $h_i(t,x)$ are defined by these equalities. Note that our assumption that the convex hulls Q_1, Q_2 have dimension two implies that $g_1, g_2 \neq 0$.

For each t, we deduce from the bilinearity of the determinant, that there exists a polynomial $H(t,x)$ such that the toric Jacobian of F_1, F_2 can be written as $J_F^T = J_{f^\eta}^T + t^{\mu(\eta)} H(t,x)$. But, as we remarked, $J_{f^\eta}^T$ lies in the ideal $I(f_1^\eta, f_2^\eta)$. Note that if for instance $\eta_1 \neq 0$, then the power of x_1 in each monomial of F_i can be obtained from the power of t and the power of x_2, that is, we could use t and x_2 as "variables" instead. We will denote by Res^X the resultant defined over X [4]. Therefore,

$$\mathrm{Res}_{A_1,A_2,A_1+A_2}(F_1, F_2, J_F^T) = \mathrm{Res}^X_{A_1,A_2,A_1+A_2}(F_1, F_2, t^{\mu(\eta)} H_2).$$

Now, it follows from Theorem 2 that

$$\mathrm{Res}^X_{A_1,A_2,A_1+A_2}(F_1, F_2, t^{\mu(\eta)}) = \mathrm{Res}_{A_1^\eta,A_2^\eta}(f_1^\eta, f_2^\eta)^{\mu(\eta)}.$$

Setting $t = 0$ we see that $\mathrm{Res}_{A_1^\eta,A_2^\eta}(f_1^\eta, f_2^\eta)^{\mu(\eta)}$ is a factor of $\mathrm{Res}_{A_1,A_2,A_1+A_2}$ (f_1, f_2, J_f^T).

If we prove that no positive power of t divides H_2 for generic coefficients, we get the desired factorization considering all possible $\eta \in \Sigma$. To see this, first note that up to multiplying each f_i by a monomial (that is, after translation of each A_i) we can assume without loss of generality that $\nu_1^\eta = \nu_2^\eta = 0$. It follows from (6.8) that $J_f^T = 0$. The polynomials g_i in (6.10) are η-homogeneous of respective degrees $\mu_1(\eta), \mu_2(\eta)$. Assume $\mu(\eta) = \mu_1(\eta) \leq \mu_2(\eta)$. In case $\mu_1(\eta) < \mu_2(\eta)$, the coefficient of t^{μ_1} in J_F^T equals the toric Jacobian $J^T_{g_1,f_2^\eta}$ of g_1 and f_2^η, which are η-homogeneous polynomials with *different* η-degrees (equal to $\mu_1(\eta) > \nu_2 = 0$). It is easy to check that $J^T_{g_1,f_2^\eta}$ is a nonzero polynomial in the coefficients of g_1, f_2^η. In case $\mu_1 = \mu_2$, we get another term which is the toric Jacobian $J^T_{f_1^\eta,g_2}$ of f_1^η and g_2, which is nonzero by the same arguments and depends on different coefficients than $J^T_{g_1,f_2^\eta}$. Thus, their sum is not the zero polynomial, as wanted.

Theorem 3 and the proof will be extended to the general n-variate setting in a forthcoming paper [7]. We only state here the following general version without proof. Recall that a lattice polytope P of dimension n in $\mathbb{R}^n$ is said to be *smooth* if at each every vertex there are n concurrent facets and their primitive inner normal directions form a basis of $\mathbb{Z}^n$. In particular, integer dilates of the unit simplex or the unit (hyper)cube are smooth.

Theorem 4 *Let $P \subset \mathbb{R}^n$ be a smooth lattice polytope of dimension n. Let $A_i = (d_i P) \cap \mathbb{Z}^n$, $i = 1, \dots, n$, $d_1, \dots, d_n \in \mathbb{Z}_{>0}$, and $f_1, \dots, f_n$ polynomials with these supports, respectively. Then, we have the following factorization*

$$\mathrm{Res}_{A_1,\dots,A_n,A_1+\cdots+A_n}(f_1, \dots, f_n, J_f^T) = \Delta_{A_1,\dots,A_n}(f_1, \dots, f_n) \cdot E,$$

where the factor E equals the finite product:

$$E = \prod_{\eta \in \Sigma} \mathrm{Res}_{A_1^\eta,\dots,A_n^\eta}(f_1^\eta, \dots, f_n^\eta).$$

Note that all the exponents in E equal 1 and all the lattice indices equal 1.

When the given lattice configurations A_i are the lattice points $d_i\sigma$ of the d_i-th dilate of the standard simplex σ in $\mathbb{R}^n$ (that is, in the homogeneous case studied in [2]), formula (6.4) gives, for any n in our notation:

$$\begin{aligned}&\mathrm{Res}_{d_1\sigma,\dots,d_n\sigma,\delta\sigma}(f_1, \dots, f_n, J_i) = \\ &\Delta_{d_1\sigma,\dots,d_n\sigma}(f_1, \dots, f_n) \cdot \mathrm{Res}_{(d_1\sigma)^{e_i},\dots,(d_n\sigma)^{e_i}}(f_1^{e_i}, \dots, f_n^{e_i}),\end{aligned}$$

where $e_0, \dots, e_n$ are the canonical basis vectors (or $e_0 = -e_1 - \cdots - e_n$, if we consider the corresponding dehomogenized polynomials, by setting $x_0 = 1$). Note that Theorem 4 gives the following more symmetric formula:

Corollary 5 *With the previous notation, it holds:*

$$\begin{aligned}&\mathrm{Res}_{d_1\sigma,\dots,d_n\sigma,(d_1+\cdots+d_n)\sigma}(f_1, \dots, f_n, J_f^T) = \\ &\Delta_{d_1\sigma,\dots,d_n\sigma}(f_1, \dots, f_n) \cdot \prod_{i=0}^{n} \mathrm{Res}_{(d_1\sigma)^{e_i},\dots,(d_n\sigma)^{e_i}}(f_1^{e_i}, \dots, f_n^{e_i}).\end{aligned}$$

It is straightforward to deduce from this expression the degree of the homogeneous mixed discriminant, obtained independently in [1, 2, 21]. Similar formulas can be obtained, for instance, in the multihomogeneous case.

We recall the following definition from [3]. If v is a vertex of A_i, we define its *mixed multiplicity* as

$$mm_{A_1,A_2}(v) := MV(Q_1, Q_2) - MV(C_i, Q_j), \quad \{i, j\} = \{1, 2\}, \tag{6.11}$$

where $C_i = conv(A_i - \{v\})$.

Let $\Sigma' \subset \Sigma$ be the set of inner normals of $A_1 + A_2$ that cut out, or define, edges e_i^η in both Q_1, Q_2. The factorization formula in Theorem 3 can be written as follows, and allows us to recover the bidegree formulas for planar mixed discriminants in [3].

Corollary 6 *Let A_1, A_2 be two lattice configurations of dimension 2 in the plane, and let f_1, f_2 be polynomials with these respective supports. Then, the resultant* *of* f_1, f_2 *and their toric Jacobian, namely* $\mathrm{Res}_{A_1,A_2,A_1+A_2}(f_1, f_2, J_f^T)$*, factors as follows:*

$$\Delta_{A_1,A_2}(f_1, f_2) \cdot \prod_{v \text{ vertex of } A_1 \text{ or } A_2} c_v^{mm_{A_1,A_2}(v)} \cdot \prod_{\eta \in \Sigma'} \mathrm{Res}_{A_1^\eta, A_2^\eta}(f_1^\eta, f_2^\eta)^{\mu(\eta)}. \tag{6.12}$$

The bidegree (δ_1, δ_2) *of the mixed discriminant* $\Delta_{A_1,A_2}(f_1, f_2)$ *in the coefficients of* f_1 *and* f_2*, respectively, is then given by the following:*

$$Vol(Q_j) + 2 \cdot MV(Q_1, Q_2) - \sum_{\eta \in \Sigma'} \ell(e_j^\eta) \cdot \mu(\eta) - \sum_{v \text{ vertex of } (A_i)} mm_{A_1,A_2}(v), \tag{6.13}$$

where $i \in \{1, 2\}$, $i \neq j$.

Proof. To prove equality (6.12), we need to show by Theorem 3 that the factor

$$E = \prod_{\eta \in \Sigma} \mathrm{Res}_{A_1^\eta, A_2^\eta}(f_1^\eta, f_2^\eta)^{\mu(\eta)}$$

equals the product

$$\prod_{v \text{ vertex of } A_1 \text{ or } A_2} c_v^{mm_{A_1,A_2}(v)} \cdot \prod_{\eta \in \Sigma'} \mathrm{Res}_{A_1^\eta, A_2^\eta}(f_1^\eta, f_2^\eta)^{\mu(\eta)}.$$

When $\eta \in \Sigma'$, i.e. η is a common inner normal to edges of both Q_1 and Q_2, we get the same factor on both terms, since that our quantity $\mu(\eta)$ coincides with the index $\min\{u(e_1(\eta), A_1), u(e_2(\eta), A_2)\}$, in the notation of [3].

Assume then that η is only an inner normal to Q_2. So, A_1^η is a vertex v, $f_1^\eta = cx^v$ is a monomial (with coefficient c) and f_2^η is a polynomial whose support equals the edge e_2^η of A_2 orthogonal to η. In this case, $\mathrm{Res}_{A_1^\eta, A_2^\eta}(f_1^\eta, f_2^\eta) = c^{\ell(f_\eta)}$ by Remark 6.2.

For such a vertex v, denote by $\boldsymbol{\mathcal{E}}(v)$ the set of those $\eta' \notin \Sigma'$ for which $v + e_2^{\eta'}$ is an edge of $Q_1 + Q_2$. Note that it follows from the proof of [3, Prop.3.13] (cf. in particular Figure 1 there), that there exist non negative integers $\mu'(\eta')$ such that

$$\mathrm{mm}(v) = \sum_{\eta' \in \boldsymbol{\mathcal{E}}(v)} \ell(e_2^{\eta'}) \cdot \mu'(\eta').$$

Indeed, $\mu(\eta') = \mu'(\eta')$.

To compute the bidegree, we use the multilinearity of the mixed volume with respect to Minkowski sum. Observe that the toric Jacobian has bidegree $(1, 1)$ in the coefficients of f_1, f_2, from which we get that the bidegree of the resultant $\mathrm{Res}_{A_1,A_2,A_1+A_2}(f_1, f_2, J_f^T)$ is equal to

$$(2MV(A_1, A_2) + \mathrm{Vol}(Q_2),\ 2MV(A_1, A_2) + \mathrm{Vol}(Q_1)). \tag{6.14}$$

Substracting the degree of the other factors and taking into account that the bidegree of the resultant $\mathrm{Res}_{A_1^\eta,A_2^\eta}(f_1^\eta, f_2^\eta)$ equals $(\ell(e_2^\eta), \ell(e_1^\eta))$, we deduce the formula (6.13), as desired.

6.4 The Multiplicativity of the Mixed Discriminant

This section studies the factorization of the discriminant when one of the polynomials factors. We make the hypothesis that f_1', f_1'', f_2 have fixed support sets $A_1', A_1'', A_2 \subseteq \mathbb{Z}^2$. So $f_1 = f_1' \cdot f_1''$ has support in the Minkowski sum $A_1 := A_1' + A_1''$; in fact, its support is generically equal to A_1. We will denote by $\mu'(\eta)$ (resp. $\mu''(\eta)$) the integer defined in (6.7), with A_1 replaced by A_1' (resp. A_1'').

Corollary 7 *Assume A_1', A_1'' and A_2 are full planar configurations of dimension 2. Let f_1', f_1'', f_2 be generic polynomials with these supports and let $f_1 = f_1' \cdot f_1''$. Then,*

$$\Delta_{A_1,A_2}(f_1, f_2) = \Delta_{A_1',A_2}(f_1', f_2) \cdot \Delta_{A_1'',A_2}(f_1'', f_2) \cdot \mathrm{Res}_{A_1',A_1'',A_2}(f_1', f_1'', f_2)^2 \cdot E,$$

where E equals the following product:

$$\prod_{\eta\in\Sigma} \mathrm{Res}_{(A_1')^\eta,A_2^\eta}((f_1')^\eta, f_2^\eta)^{\mu'(\eta)-\mu(\eta)} \cdot \mathrm{Res}_{(A_1'')^\eta,A_2^\eta}((f_1'')^\eta, f_2^\eta)^{\mu''(\eta)-\mu(\eta)}. \tag{6.15}$$

Proof. By Theorem 3, we get that

$$\Delta_{A_1,A_2}(f_1, f_2) = \frac{\mathrm{Res}_{A_1,A_2,A_1+A_2}(f_1, f_2, J_f^T)}{\prod_{\eta\in\Sigma} \mathrm{Res}_{A_1^\eta,A_2^\eta}(f_1^\eta, f_2^\eta)^{\mu(\eta)}}, \tag{6.16}$$

and similarly for $\Delta_{A_1',A_2}(f_1', f_2)$ and $\Delta_{A_1'',A_2}(f_1'', f_2)$. Let us write the numerator of (6.16) as follows:

$$\mathrm{Res}_{A_1'+A_1'',A_2,A_1'+A_1''+A_2}(f_1' f_1'', f_2, J^T_{f_1' f_1'', f_2}),$$

where $J^T_{f_1' f_1'', f_2} = f_1' J^T_{f_1'' f_2} + f_1'' J^T_{f_1', f_2}$. We now apply Theorem 2 to re-write it as follows:

$$\mathrm{Res}_{A_1',A_2,A_1'+A_1''+A_2}(f_1', f_2, J^T_{f_1' f_1'', f_2})\,\mathrm{Res}_{A_1'',A_2,A_1'+A_1''+A_2}(f_1'', f_2, J^T_{f_1' f_1'', f_2})$$
$$= \mathrm{Res}_{A_1',A_2,A_1'+A_1''+A_2}(f_1', f_2, f_1'' J^T_{f_1', f_2})\,\mathrm{Res}_{A_1'',A_2,A_1'+A_1''+A_2}(f_1'', f_2, f_1' J^T_{f_1'', f_2}),$$

because the resultant of $\{h_1, h_2 + g h_1, \ldots\}$ equals the resultant of $\{h_1, h_2, \ldots\}$, for any choice of polynomials h_1, h_2, g (with suitable supports). We employ again

Theorem 2 to finalize the numerator as follows:

$$\mathrm{Res}_{A_1',A_2,A_1'+A_2}(f_1', f_2, J^T_{f_1',f_2}) \cdot \mathrm{Res}_{A_1'',A_2,A_1''+A_2}(f_1'', f_2, J^T_{f_1'',f_2}) \cdot \mathrm{Res}_{A_1',A_1'',A_2}(f_1', f_1'', f_2)^2.$$

For the denominator of (6.16), we use again Theorem 2 to write:

$$\prod_{\eta\in\Sigma'} \mathrm{Res}_{A_1'^{\eta},A_2^{\eta}}(f_1'^{\eta}, f_2^{\eta})^{\mu'(\eta)} \cdot \prod_{\eta\in\Sigma''} \mathrm{Res}_{A_1''^{\eta},A_2^{\eta}}(f_1''^{\eta}, f_2^{\eta})^{\mu''(\eta)} = \prod_{\eta\in\Sigma} \mathrm{Res}_{A_1'^{\eta}+A_1''^{\eta},A_2^{\eta}}(f'^{\eta}_1 f''^{\eta}_1, f_2^{\eta})^{\mu(\eta)} \cdot E,$$

because the products

$$\prod_{\eta\in\Sigma\setminus\Sigma'} \mathrm{Res}_{A_1'^{\eta},A_2^{\eta}}(f_1'^{\eta}, f_2^{\eta})^{\mu'(\eta)} = \prod_{\eta\in\Sigma\setminus\Sigma''} \mathrm{Res}_{A_1''^{\eta},A_2^{\eta}}(f_1''^{\eta}, f_2^{\eta})^{\mu''(\eta)} = 1,$$

since $f_1'^{\eta}, f_2^{\eta}$ (resp. $f_1''^{\eta}, f_2^{\eta}$) are both monomials. To conclude the proof, simply assemble the above equations.

As a consequence, we have $\deg_{A_1,A_2} \Delta(f_1, f_2) =$

$$= \deg_{A_1',A_2} \Delta(f_1', f_2) + \deg_{A_1'',A_2} \Delta(f_1'', f_2) + 2 \cdot \deg_{A_1',A_1'',A_2} \mathrm{Res}(f_1', f_1'', f_2) - \deg(E).$$

When all the configurations are full and with the same normal fan, all the exponents $\mu(\eta) = \mu'(\eta) = \mu''(\eta) = 1$. Therefore, $E = 1$ and no extra factor occurs.

We define $\mu_1'(\eta), \mu_1''(\eta)$ as in (6.6). Indeed, we now fix η and will simply write $\mu_1', \mu_1'', \mu_1, \mu_2$. It happens that only one of the factors associated to η can occur in E with non zero coefficient. More explicitly, we have the following corollary, whose proof is straightforward.

Corollary 8 *With the notations of Corollary 7, for any $\eta \in \Sigma$ it holds that:*

- *If $\mu_1' = \mu_1''$, then $\mu' = \mu'' = \mu$ and there is no factor in E "coming from η".*
- *If $\mu_1' \neq \mu_1''$, assume wlog that $\mu_1 = \mu_1' < \mu_1''$. There are three subcases:*
  - *If $\mu_2 \leq \mu_1$, again there is no factor in E "coming from η".*
  - *If $\mu_1 = \mu_1' < \mu_2 < \mu_1''$, then the resultant $\mathrm{Res}_{(A_1')^{\eta},A_2^{\eta}}((f_1')^{\eta}, f_2^{\eta})$ does not occur, but $\mathrm{Res}_{(A_1'')^{\eta},A_2^{\eta}}((f_1'')^{\eta}, f_2^{\eta})$ has nonzero exponent (this resultant could just be the coefficient of a vertex raised to the mixed multiplicity).*
  - *If $\mu_1 = \mu_1' < \mu_1'' \leq \mu_2$, the situation is just the opposite than in the previous case.*

Conclusion and Future Work

The intent of this book chapter was to present our main results relating the mixed discriminant of two bivariate Laurent polynomials with fixed support, with the sparse resultant of two bivariate Laurent polynomials with fixed support and their toric Jacobian. On our way, we deduced a general multiplicativity formula for the mixed discriminant when one polynomial factors as $f = f' \cdot f''$. This formula occurred as a consequence of our main result, Theorem 3, and generalized known formulas in the homogeneous case to the sparse setting. Furthermore, we obtained a new proof of the bidegree formula for planar mixed discriminants, which appeared in [3].

The generalization of our formulas to any number of variables will allow us to extend our applications and to develop effective computational techniques for sparse discriminants based on software for the computation of resultants.

Acknowledgements A. Dickenstein is partially supported by UBACYT 20020100100242, CONICET PIP 11220110100580 and ANPCyT 2008-0902 (Argentina). She also acknowledges the support of the M. Curie Initial Training Network "SAGA" (ShApes, Geometry, and Algebra; FP7-PEOPLE contract PITN-GA-2008-214584), which made possible her visit to the University of Athens in January 2012, where this work was initiated. A. Karasoulou's research has received funding from the European Union (European Social Fund) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework, Research Funding Program "ARISTEIA", Project ESPRESSO: Exploiting Structure in Polynomial Equation and System Solving in Geometric and Game Modeling.

References

1. O. Benoist, Degrés d'homogénéité de l'ensemble des intersactions complètes singulières. Annales de l'Institut Fourier **62**(3), 1189–1214 (2012)
2. L. Busé, J.-P. Jouanolou, On the discriminant scheme of homogeneous polynomials. Math. Comput. Sci. **2**(8), 175–234 (2014)
3. E. Cattani, M.A. Cueto, A. Dickenstein, S. Di Rocco, B. Sturmfels, Mixed discriminants. Math. Z. **274**, 761–778 (2013)
4. E. Cattani, A. Dickenstein, B. Sturmfels, Residues and resultants. J. Math. Sci. **5**, 119–148 (1998). University of Tokyo
5. C. D'Andrea, M. Sombra, A Poisson formula for the sparse resultant (2013, preprint). arXiv:1310.6617
6. A. Dickenstein, A world of binomials, in *Foundations of Computational Mathematics*, Hong Kong, 2008, ed. by F. Cucker, A. Pinkus, M. Todd. London Mathematical Society Lecture Note Series, vol. 363 (Cambridge University Press, Cambridge/New York, 2009), pp. 42–66
7. A. Dickenstein, A. Esterov, Mixed discriminants and toric jacobians (2014, manuscript in progress)
8. A. Dickenstein, E.M. Feichtner, B. Sturmfels, Tropical discriminants. J. Am. Math. Soc. **20**, 1111–1133 (2007)
9. A. Dickenstein, J.M. Rojas, K. Rusek, J. Shih, Extremal real algebraic geometry and A-discriminants. Mosc. Math. J. **7**(3), 425–452 (2007)

10. A. Dickenstein, L.F. Tabera, Singular tropical hypersurfaces. Discret. Comput. Geom. **47**(2), 430–453 (2012)
11. I.Z. Emiris, V. Fisikopoulos, C. Konaxis, L. Penaranda, An Oracle-Based, Output-Sensitive algorithm for projections of resultant polytopes. Int. J. Comput. Geom. Appl. Spec. Issue **23**(4), 397–423 (2014)
12. I.Z. Emiris, T. Kalinka, C. Konaxis, T. Luu Ba, Sparse implicitization by interpolation: characterizing non-exactness and an application to computing discriminants. J. Comput. Aided Des. Spec. Issue **45**, 252–261 (2013)
13. I.Z. Emiris, E. Tsigaridas, G. Tzoumas, The predicates for the exact Voronoi diagram of ellipses under the Euclidean metric. Int. J. Comput. Geom. Appl. **18**(6), 567–597 (2008)
14. A. Esterov, Newton polyhedra of discriminants of projections. Discret. Comput. Geom. **44**(1), 96–148 (2010)
15. A. Esterov, The discriminant of a system of equations. Adv. Math. **245**, 534–572 (2013)
16. J.-C. Faugére, G. Moroz, F. Rouillier, M. Safey El Din, Classification of the perspective-three-point problem: discriminant variety and real solving polynomial systems of inequalities, in *Proceedings of ACM ISSAC*, Hagenberg (2008), pp. 79–86
17. I.M. Gel'fand, M.M. Kapranov, A.V. Zelevinsky, Hypergeometric functions and toric varieties. Funktsional. Anal. i Prilozhen. **23**(2), 12–26 (1989)
18. I.M. Gel'fand, M.M. Kapranov, A.V. Zelevinsky, *Discriminants, Resultants and Multidimensional Determinants* (Birkhäuser, Boston, 1994)
19. L. González-Vega, I. Nacula, Efficient topology determination of implicitly defined algebraic plane curves. Comput. Aided Geom. Des. **19**(9), 719–743 (2002)
20. G. Moroz, F. Rouiller, D. Chablat, P. Wenger, On the determination of cusp points of 3-RPR parallel manipulators. Mech. Mach. Theory **45**(11), 1555–1567 (2010)
21. J. Nie, Discriminants and non-negative polynomials. J. Symb. Comput. **47**, 167–191 (2012)
22. P. Pedersen, B. Sturmfels, Product formulas for resultants and Chow forms. Math. Z. **214**, 377–396 (1993)
23. E.F. Rincón, Computing tropical linear spaces. J. Symb. Comput. **51**, 86–93 (2013)
24. M. Shub, S. Smale, Complexity of Bézout's theorem I. Geometric aspects. J. Am. Math. Soc. **6**(2), 459–501 (1993)
25. B. Sturmfels, The Newton polytope of the resultant. J. Algebr. Comb. **3**, 207–236 (1994)
26. J.J. Sylvester, Sur l'extension de la théorie des résultants algébriques. Comptes Rendus de l'Académie des Sciences **LVIII**, 1074–1079 (1864)

Chapter 7
Topology of the Intersection of Two Parameterized Surfaces, Using Computations in 4D Space

Stéphane Chau and André Galligo

7.1 Introduction

7.1.1 Interest of the Problem

In Computer Aided Geometric Design (CAGD), parameterized surfaces are used to delimit volumes. The computation of the intersection curve between two such surfaces is thus crucial for the description of CAGD objects. A simple method to address this problem consists in using a mesh for each surface, and then proceed to their intersection via intersection of triangles. But the instability created by intersecting almost parallel triangles constitutes a drawback. A more stable method relies on global representations of the surfaces by B-splines; however the usual CAGD procedures (offsetting, drafting, ...) do not keep this model. In practice, so-called procedural surfaces (i.e. given by evaluation) are used, in CAGD systems, to represent sequences of constructions indicated by the user. Then a B-spline approximation is computed for further developments. So, even if the intersection method is exact, in its final step, it only provides an approximation of the "real" intersection curve.

Ideally, approximations of the surfaces should not be separated from the intersection process. An intermediate strategy is to approximate the given surfaces by meshes of algebraic shapes more complex than the triangles; hence the intersection

Corresponding author and member of the Galaad project team, INRIA-Sophia Antipolis.

S. Chau • A. Galligo (✉)
Laboratoire de Mathématiques, Université de Nice Sophia Antipolis, Parc Valrose, 06108 Nice Cedex 2, France
e-mail: Andre.GALLIGO@unice.fr

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_7

locus will be more precise. A good choice is to approximate by Bézier surface patches of small degree (see Sect. 7.1.2). Then, it is crucial to be able to efficiently intersect two such polynomial parameterized surfaces.

The present paper aims to contribute to a robust solution of this problem which avoids some drawbacks such as large intermediate algebraic expressions that appear in projection methods.

The intersection curve of two such parameterized surfaces is characterized by 3 equations $F_i(s,t) = G_i(u,v)$, $i = 1,2,3$ of 4 variables. So, it is the image of a curve in a four dimensional space. We provide a method to draw this curve with a guaranteed topology.

Our method depends ultimately on an a priori fixed threshold epsilon, which controls the approximations. In practice, this value is determined by the engineers for the targeted applications: in CAD-CAM a precision of 1 μ is often required.

7.1.2 An Example of Biquadratic Meshing of a Procedural Surface

Let $\mathcal{S}$ be a general parameterized surface given by evaluations. We consider a grid of points on $\mathcal{S}$ of size $(2m+1, 2n+1)$. This is used to build a grid of biquadratic patches of size (m,n). Figure 7.1 (left) illustrates this grid in the 2D parameter space. Thus, the coefficients are shared between adjacent patches. An example of this kind of approximation is given in Fig. 7.2. In this example, we have a shape composed by three B-spline surfaces on the left, then we consider an offset, which cannot be represented by a B-spline, and we approximate it by a grid of 144

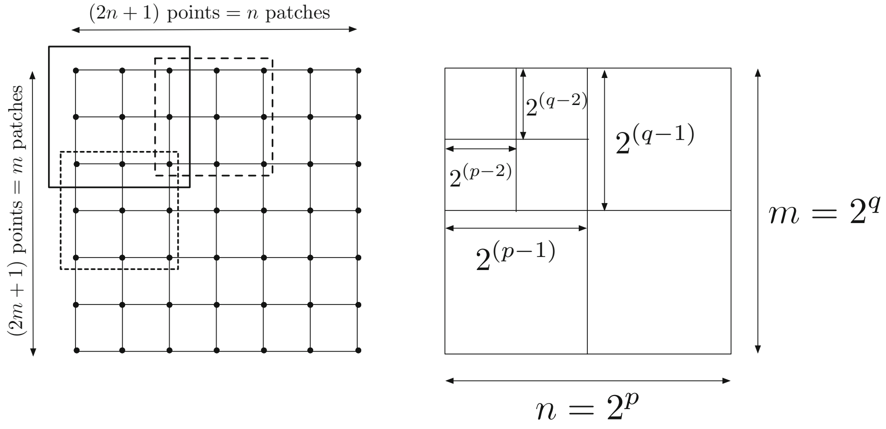


Fig. 7.1 Grid of biquadratic patches on the *left*. Grid of boxes with $n = 2^p$ and $m = 2^q$ on the *right*

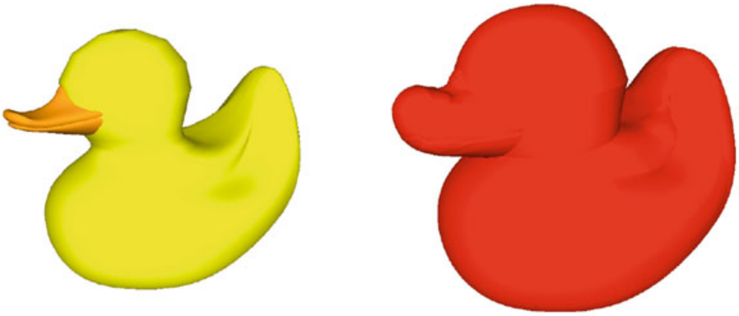


Fig. 7.2 Approximation of an offset by a grid of 144 biquadratic patches

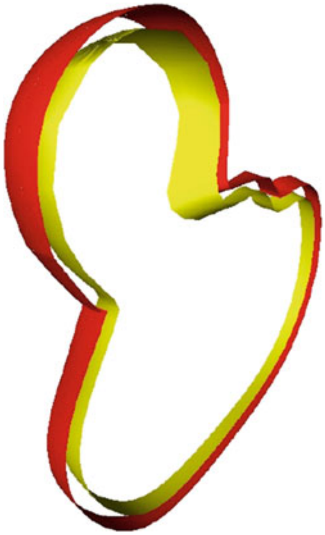


Fig. 7.3 Approximation of an offset by a grid of 144 biquadratic patches (clipped picture)

biquadratic patches (the result is shown on the right). In order to provide a better view of the offset, a clipped picture is also given (Fig. 7.3).

Now, we consider two such grids and hierarchies on $\mathcal{S}_1$ and $\mathcal{S}_2$, the two surfaces to be intersected. We produce another grid of $m \times n$ 3D boxes taking min-max values of the patch coefficients, each box containing the patch due to the convex hull property of the Bézier surfaces. Then, we build a quadtree hierarchy covering this grid. Figure 7.1 (right) illustrates this construction. Using these quadtrees we look for intersecting boxes and we obtain a set of pairs of intersecting boxes associated to patches. This process is efficient and, as we will see, provides a good description of the intersection curve. However, it requires an efficient and robust algorithm for the intersection of two Bézier surface patches.

In the sequel, we concentrate on the presentation of our subdivision algorithm for the intersection of polynomial patches.

7.1.3 Organization of the Paper

In Sect. 7.2, a brief description of previous work on topology computation is given. We sketch an introduction on the subdivision approach for plane curves. Section 7.3 is devoted to the algorithmic study of the topology of an implicit four dimensional curve. A complete description of its computation, by a subdivision method, is given in this case. In fact, this case corresponds to the intersection curve between two polynomial parameterized patches.

Implementation aspects are addressed in Sect. 7.4 and examples are presented. The last Sect. 7.5 addresses the problem of computing exactly the image of that curve in $\mathbb{R}^3$. The hardness of the problem is demonstrated, then a solution is outlined.

7.2 Previous Work on Topology Computation of a Curve

The problem of tracing the intersection of two surfaces, also called surface-surface intersection, is a classical problem in Computer Aided geometric Design, and was addressed very early in that community, and improved continuously, see e.g. [5, 12, 16, 20, 23, 27, 28] and the references therein. It was also considered rather early, together with the estimation of its complexity in the Computer algebra community, see e.g. [4, 7, 14, 22].

The contributions and the progresses increased in the last decade. One distinguishes between 2D and 3D curves, between the cases of implicit or parametric curves and also between algebraic or subdivision methods which proceed up to a predefined precision.

7.2.1 Isotopic Curve

The topology of an algebraic curve $\mathcal{C}$ in $\mathbb{R}^n$ $(n \geq 2)$ can be represented by a list of line segments whose concatenation forms a curve isotopic to $\mathcal{C}$. In the last decade, several algorithmic papers started the study of this subject, see e.g. [13, 18, 21, 25] and their bibliography. They use the following basic constructions to make this mathematical definition effective. Sweeping methods rely on parallel lines or planes and detect topological events (critical points) such as we will see, tangent points to the sweeping planes or singularities; we refer to [19,20] for planar curves and [3,18] for spatial curves. With these algebraic approaches, the precise determination of the critical points generally requires to compute sub-resultant sequences and is often time consuming. This and the computation of the complexity has been improved recently with the works of [1, 2, 9–11, 15, 24]; several nice complexity results are now available.

Subdivision and exclusion techniques, see [25, 26], rely on (simple) criteria to remove unnecessary domains, then restrict to domains where the situation is tame. Polynomial representation in Bernstein bases is generally preferred, see [16, 27].

7.2.2 *Regularity Test and Subdivision Method, the Case of a Plane Curve*

A subdivision approach to compute the topology of the intersection curve between two algebraic surfaces is given in [25]. It consists in subdividing the domain until a regularity test is satisfied. Let us briefly recall it. To simplify the presentation, we consider in this subsection the case of a plane curve.

Let $f(x, y)$ be a polynomial and $B = [a, b] \times [c, d] \subset \mathbb{R}^2$ a box. Consider the implicit plane curve associated to f in the box B by the equation $f(x, y) = 0$. A regularity test allows to determine uniquely the topology of the curve in the box from its intersection with the boundary. A collection of segments is provided, which realizes an isotopy.

Proposition 7.1 *If* $\partial_y f(x, y) \neq 0$ *for all* $(x, y) \in B = [a, b] \times [c, d]$*, then for all* $x \in [a, b]$ *there exists at most one* $y \in [c, d]$ *such that* $f(x, y) = 0$.

Proof: Let x_0 be a value in $[a, b]$. If there were two different values $y_0 < y_1$ in $[c, d]$ such that $f(x_0, y_0) = f(x_0, y_1) = 0$ then by Rolle's theorem, there would exist $y_2 \in [y_0, y_1]$ such that $\partial_y f(x_0, y_2) = 0$. ■

Remark 7.1
- This criterion considers $\partial_y f(x, y)$ for all values (x, y) in the box and not only for all points of the curve, so it is rather restrictive.
- To implement this criterion, the polynomial $\partial_y f(x, y)$ is expressed in Bernstein basis and the coefficients are required to share the same sign.
- A similar statement holds replacing the condition $\partial_y f(x, y) \neq 0$ by $\partial_x f(x, y) \neq 0$ (for all $(x, y) \in B$).

If f satisfies this test, then the topology of the curve $\{(x, y) \in B \mid f(x, y) = 0\}$ can be determined by knowing the intersection points between the curve and the border of B. Hence, a first step is to compute all these intersection points (a point is repeated if its multiplicity is even) and sort them by their x component to obtain a list of points $p_1, p_2, \ldots, p_{2s-1}, p_{2s}$. Then, in the box, the curve is isotopic to the set of segments: $[p_1, p_2], \ldots, [p_{2s-1}, p_{2s}]$ (see the illustration in Fig. 7.4). The criterion can be checked recursively by subdividing the initial curve (using De Casteljau's algorithm) until a family of boxes is obtained where the test is verified. The approach is extended (in [25]) to the case of a 3D curve defined implicitly by two equations. This provides an elegant and efficient solution to the topology computation problem of an intersection curve between two implicit surfaces.

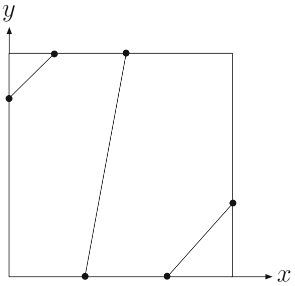


Fig. 7.4 Topology via regularity test in 2D case

7.2.3 No Loop and Single Component

The results in the present work are part of the PhD thesis of the first author [8].

In our review of the previous works, we point out the interesting recent paper [6] which approach is close to our method, but developed independently. Paper [6] is also based on a subdivision method and the use of tests to guaranty that “there is just one component [of the intersection] over a given domain”, this property is similar to what we described in the previous subsection and to our generalized injectivity requirements (see Sect. 7.3.2). However the formalisms are different. Indeed, in that work, the authors rely on (more general) wedge products and Gaussian hyperspheres; while here all the intermediate constructions are elementary and the case by case descriptions are simple and adapted to our precise target. Moreover, [6] does not consider the knot problem that we addressed in the our section.

7.3 Topology of (Parameterized) Surface/Surface Intersection

7.3.1 Equations

This is the main technical section, where our setting, our intermediate constructions and our criteria are presented.

Let **F** and **G** be two polynomial surface patches defined by the maps:

$$F : \begin{pmatrix} [0,1]^2 \longrightarrow \mathbb{R}^3 \\ (s,t) \longmapsto F(s,t) \end{pmatrix}$$

$$G : \begin{pmatrix} [0,1]^2 \longrightarrow \mathbb{R}^3 \\ (u,v) \longmapsto G(u,v) \end{pmatrix}.$$

We suppose that the intersection $\mathbf{F} \cap \mathbf{G}$ is a curve $\mathbf{C}$ in $\mathbb{R}^3$; it is the image of

$$\mathcal{C} = \left\{(s,t,u,v) \in [0,1]^4 \mid F(s,t) - G(u,v) = 0\right\}.$$

Our aim is to compute the topology of $\mathcal{C}$ by a subdivision method generalizing the approach described in Sect. 7.2. An injectivity criterion which says that for all $s_0 \in [0,1]$, there exists at most one $(t_0, u_0, v_0) \in [0,1]^3$ such that $F(s_0,t_0) - G(u_0,v_0) = 0$, is needed. So, we fix $s_0 \in [0,1]$, and study the map:

$$\phi : \begin{pmatrix} [0,1]^3 & \longrightarrow \mathbb{R}^3 \\ (t,u,v) & \longmapsto F(s_0,t) - G(u,v) \end{pmatrix}$$

Thereafter, we set $\phi(t,u,v) = F(s_0,t) - G(u,v) = (\phi_1, \phi_2, \phi_3)$.

7.3.2 Topology of a 4 Dimensional Implicit Curve (Regularity Criterion)

Let us first recall basic facts and definitions.

Consider a differential map $f : E_1 \to E_2$. Then, f is not injective, if there exists a "witness" pair $x_1, x_2 \in E_1$ such that $f(x_1) = f(x_2)$. We also say that f is (resp. is not) locally injective at a point x_0, if the jacobian determinant of f at x_0 is invertible (resp. vanishes)

Computationally, one distinguishes two cases of non injectivity. First, it is a local behavior, i.e. there exists a point $x_0 \in E_1$ such that in any (small) neighborhood of x_0 there exist such a witness pair x_1, x_2 then, since f is a differential map, this phenomena can be detected on a linear approximation of f at x_0, by computing the jacobian determinant of f, hence relying on local non injectivity. Second, the witness pair (x_1, x_2) of defect of injectivity is such that x_1 and x_2 are far apart, then the defect of injectivity is a priori difficult to control computationally; a usual strategy is to connect the points x_1 and x_2 along a curve, in order to detect another witness point $x_0 \in E_1$ where f is not locally injective.

Construction of the Injectivity Criterion for ϕ

A necessary condition of injectivity is the local injectivity of ϕ. By the inverse function theorem, it is satisfied when the jacobian of ϕ is nonzero over $[0,1]^3$:

$$\forall (t,u,v) \in [0,1]^3, \det\left(\partial_t \phi(t,u,v), \partial_u \phi(t,u,v), \partial_v \phi(t,u,v)\right) \neq 0.$$

Now, we assume local injectivity, and we look for sufficient conditions of (global) injectivity of ϕ.

If ϕ is not injective, there exist two different points A and B in $[0, 1]^3$ such that for all $i \in \{1, 2, 3\}$, $\phi_i(A) = \phi_i(B)$. For our analysis, we fix A and we introduce the two following subsets of $[0, 1]^3$, which will be useful to connect A and B:

$$S_1 := \left\{M \in [0, 1]^3 \mid \phi_1(M) = \phi_1(A)\right\} \tag{7.1}$$

$$C_{1,2} := \left\{M \in [0, 1]^3 \mid \phi_1(M) = \phi_1(A) \text{ and } \phi_2(M) = \phi_2(A)\right\}. \tag{7.2}$$

First case: Let us rule out the possibility that A and B are on a same connected component of $C_{1,2}$ that we denote by Γ. Since Γ is a connected curve, local injectivity of ϕ, and the inverse function theorem, implies that Γ can be differentiably parameterized. Then, ϕ_3 restricted to Γ is differentiable and takes the same value at A and B; by Rolle's theorem ϕ_3 admits an extremum on $C_{1,2}$. This implies that ϕ is not locally injective near the argument of the extremum; a contradiction with our assumption of local injectivity.

Second case: A and B are on two different connected components of $C_{1,2}$ denoted by C_A and C_B. One of these two curves describes a loop. This is ruled out with exactly the same reasoning as in the First case.

Third case: A and B are on two different connected components of $C_{1,2}$ denoted by C_A and C_B. None of these two curves describes a loop.

Hence, C_A (as well as C_B) intersects twice the border of the cube $[0, 1]^3$. Then, $C_{1,2}$ intersects the border of the cube $[0, 1]^3$ in four distinct points.Therefore, if we can rule out this last possibility, we will get a sufficient condition of injectivity.

To prevent the possibility of several intersection points of $C_{1,2}$ with the border, our strategy is to impose sufficient monotony conditions on ϕ_1 and ϕ_2. We are guided by the fact that at all these intersection points ϕ_1 and ϕ_2 take the same values, since they belong to $C_{1,2}$. Later, a subdivision process will monitor these monotony properties.

Monotony Condition on ϕ_1

First, we impose monotony conditions on ϕ_1 restricted to the edges of $[0, 1]^3$. For example, we can require that ϕ_1 increases on each edge of $[0, 1]^3$ as indicated in Fig. 7.5. So $\phi_1 - \phi_1(A)$ vanishes at most once on each path going from the vertex O to the vertex I following the ordered edges. This condition implies that the implicit surface S_1 (of equation $\phi_1(t, u, v) = \phi_1(A)$) is connected. Indeed, if S_1 admitted two connected components in the cube, they would intersect the edges at the same points and contradict differentiability of ϕ, hence it does not happen.

We classify all possible configurations by the number of the intersection points (3, 4, 5 or 6) between S_1 and the edges, as illustrated in Fig. 7.6. Note that since $C_{1,2} \subset S_1$ and $\partial S_1 \subset \partial[0, 1]^3$, the equality $\#(C_{1,2} \cap \partial S_1) = \#(C_{1,2} \cap \partial[0, 1]^3)$ holds (see Fig. 7.7).

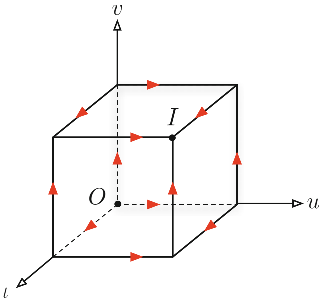


Fig. 7.5 Example of monotony of ϕ_1 on the edges of $[0, 1]^3$

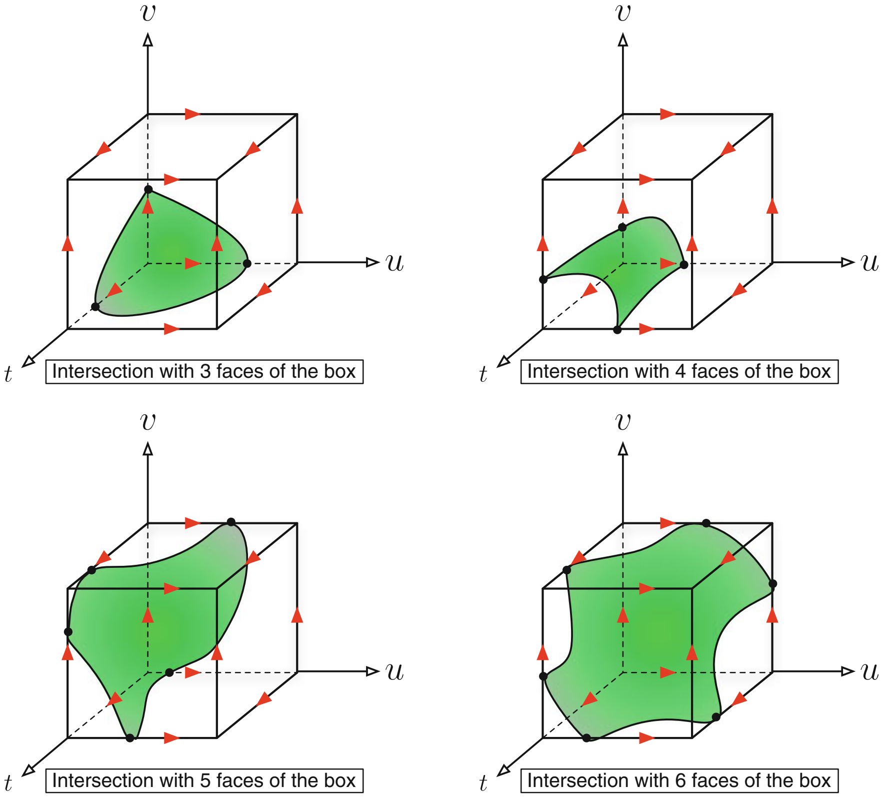

v
u
t
Intersection with 3 faces of the box
v
u
t
Intersection with 4 faces of the box
v
u
t
Intersection with 5 faces of the box
v
u
t
Intersection with 6 faces of the box

Fig. 7.6 Configurations of the surface S_1 in $[0, 1]^3$ under the monotony constraint on ϕ_1

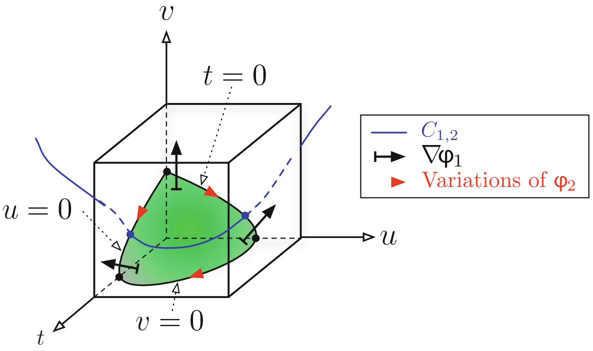


Fig. 7.7 Example of monotony of ϕ_2 along the border of S_1

Monotony Condition on ϕ_2 Along ∂S_1

Now, we study each configuration. We impose monotony conditions on ϕ_2 along the border of S_1 to force $C_{1,2}$ to have at most two intersection points with this border. Hence contradicts the possibility of having four intersection points with the border of the cube.

The following lemma will be useful:

Lemma 7.1 *Let f be a $\mathcal{C}^1$ real function over an open convex set $\mathcal{U} \subset \mathbb{R}^2$ and h be a nonzero vector in $\mathbb{R}^2$. If for all $u \in \mathcal{U}$ we have $\nabla f(u) \cdot h > 0$, then f is increasing in the direction h on $\mathcal{U}$ i.e. $\forall u \in \mathcal{U}$ and $\forall \epsilon > 0$ such that $(u + \epsilon\, h) \in \mathcal{U}$, we have $f(u + \epsilon\, h) > f(u)$.*

Proof: Let $u_0 \in \mathcal{U}$ and $\epsilon > 0$ such that $(u + \epsilon\, h) \in \mathcal{U}$. Then $f(u_0 + \epsilon\, h) - f(u_0) = \int_0^1 \varphi(t)\, dt$ with $\varphi(t) = \nabla f\ (u_0 + t\epsilon h) \cdot h$ which is positive. ■

- Replacing f by $-f$, we get similarly that $\nabla f(u) \cdot h < 0$ implies f is decreasing in the direction h on $\mathcal{U}$.
- Remember that in the plane, for a nonzero vector $\overrightarrow{w} = (a, b)$, the vector $\overrightarrow{w}^{\perp} :=$ $(-b, a)$ is normal to $\overrightarrow{w}$ and the oriented angle $\widehat{(\overrightarrow{w}, \overrightarrow{w}^{\perp})}$ is equal to $\pi/2$.

So, in order to ensure monotony of ϕ_2 along the border of S_1 (see Fig. 7.7), we orient S_1 by the vector field $\nabla\phi_1$. This induces an orientation on the border ∂S_1 of S_1; ∂S_1 is the intersection of S_1 with the faces of the cube. This orientation of the border of S_1 in each face is given by $\overrightarrow{w}^{\perp}$ where $\overrightarrow{w}$ is the projection of $\nabla\phi_1$ on the faces. Then, we impose a monotony direction of ϕ_2 restricted to ∂S_1 on each face of the cube. To illustrate this procedure, Fig. 7.8 represents, in the three coordinates planes, the monotonies shown on Fig. 7.7: the desired monotony in the (u, v)-plane (pictured in the middle of Fig. 7.8) is obtained by projecting the vector $\nabla\phi_1$ on this

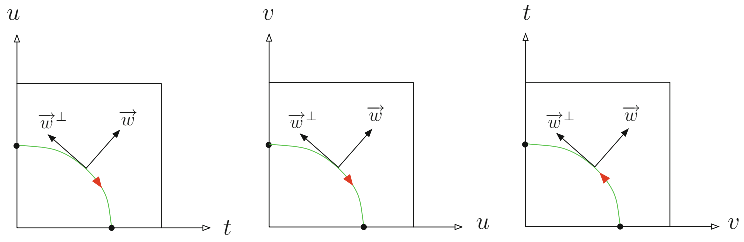


Fig. 7.8 Traces of S_1 on the faces of the box with orientations

plane and we have $\vec{w} = (\partial_u \phi_1(0,\cdot), \partial_v \phi_1(0,\cdot))$. Then, we force the decreasing of $(u,v) \mapsto \phi_2(0,u,v)$ in the direction $\vec{w}^{\perp}$. Applying Lemma 7.1, we require:

$$\forall (u,v) \in [0,1]^2, \begin{pmatrix} \partial_u \phi_2(0,u,v) \\ \partial_v \phi_2(0,u,v) \end{pmatrix} \cdot \begin{pmatrix} -\partial_v \phi_1(0,u,v) \\ \partial_u \phi_1(0,u,v) \end{pmatrix} < 0.$$

This dot product is a polynomial of bi-degree $(3,3)$ with respect to the variables (u,v). Considered also as a polynomial in s (that we fixed at the beginning of this section) it is of degree 4 in s.

Choice of Monotony Constraints

Here, we present our choice of sufficient condition such that $\#(C_{1,2} \cap \partial S_1) \leq 2$. First, we consider the case when S_1 intersects the six faces of the cube $[0,1]^3$. In the other cases, we just skip the condition corresponding to missing segments contracted to a point (see Figs. 7.9 and 7.10).

The border of S_1 is isotopic to an hexagon $\{M_1, \ldots, M_6\}$ as shown on Fig. 7.11.

A sufficient monotony condition is given by a choice of an initial point M_I and a final point M_F among $\{M_1, \ldots, M_6\}$ with the possible choice $M_I = M_F$ such that ϕ_2 is monotonic on the paths on ∂S_1 joining M_I to M_F. This clearly implies that $\phi_2 - \phi_2(A)$ vanishes at most twice on ∂S_1. Since the four variables $\{s,t,u,v\}$ play similar roles, we extend our choice of sufficient conditions applying permutations on these letters.

1. Instead of fixing s, we can fix t, u or v and consider the corresponding maps.
2. The roles of ϕ_1, ϕ_2, and ϕ_3 can be exchanged.

All these options will be considered to speed up the implementation.

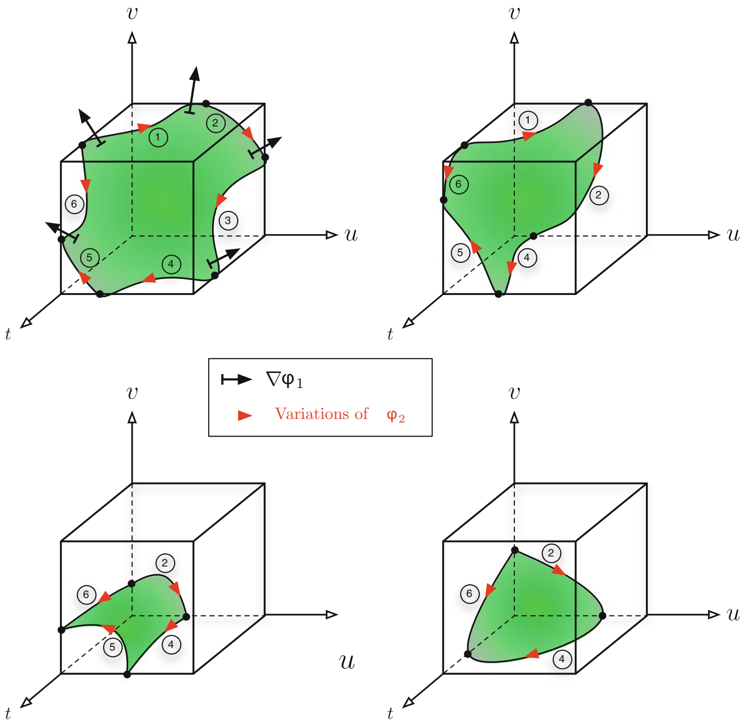


Fig. 7.9 Exemple of monotony of ϕ_2 along the border of S_1 in the case when S_1 intersects six faces of the box and the resulting configurations in the other cases

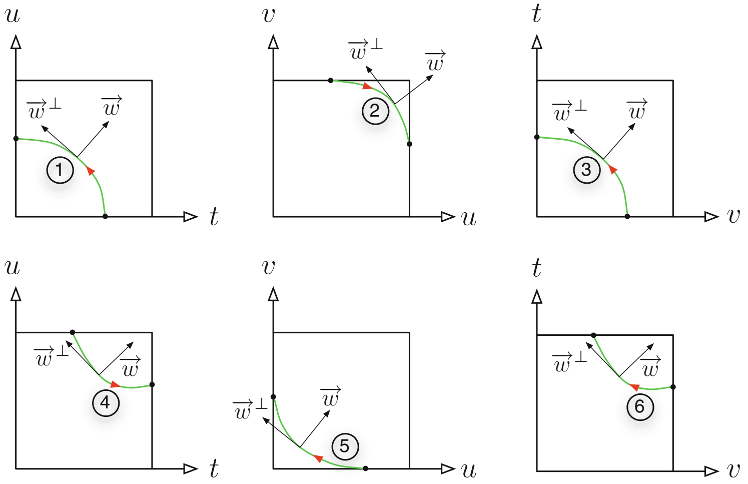


Fig. 7.10 Traces of S_1 on the faces of the box (it corresponds to the case represented in Fig. 7.9)

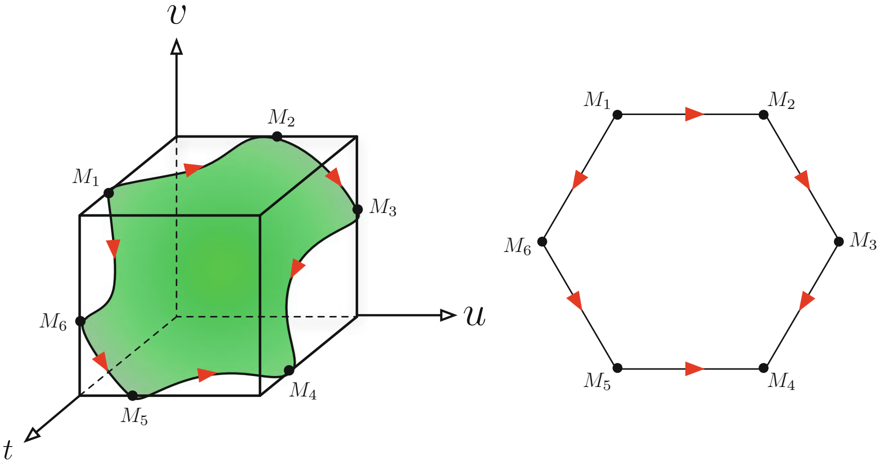


Fig. 7.11 Traces of S_1 on the faces of the box (it corresponds to the case represented in Fig. 7.9)

7.4 Algorithms and Data Structures

In this section, we present implementation aspects of our intersection algorithm we used Axel[1] an algebraic geometric modeler.

7.4.1 Hextree Data Structure and Topology

A subdivision algorithm on a box in $[a_1, b_1] \times [a_2, b_2] \times [a_3, b_3] \times [a_4, b_4] \subset \mathbb{R}^4$ explores sub-boxes constructed by considering intermediate values c_i between a_i and b_i for $i \in \{1, \ldots, 4\}$; here we choose $c_i = \dfrac{a_i + b_i}{2}$. So a box has 16 sub-boxes. Iterating this construction, an hextree is build; i.e. each node of the tree has 16 children numbered from 0 to 15. In binary expression, this number is written $\alpha_1\alpha_2\alpha_3\alpha_4$ with $\alpha_i = 0$ or 1; for $i \in \{1, \ldots, 4\}$, if $\alpha_i = 0$, the sub-box is constructed over $[a_i, c_i]$ and if $\alpha_i = 1$ it is constructed over $[c_i, b_i]$. For example, the twelfth child is written 1100 and corresponds to the sub-box $[c_1, b_1] \times [c_2, b_2] \times [a_3, c_3] \times [a_4, c_4]$.

This is called an hextree data structure, it generalizes the quadtrees which are widely used to represent planar shapes. See e.g. [17] A label which stores the needed information is associated to each node of the tree. Here, the information will be the description of the topology of the intersection curve $\mathcal{C}$ into the corresponding sub-

[1] http://axel.inria.fr

box. More precisely, we require that, at the leaves of the tree, this intersection is empty or its dimensions are below some threshold or it is isotopic to a collection of disjoint segments; each segment connects two intersection points of the curve $\mathcal{C}$ with the border of the considered sub-box. Each such segment is represented by the coordinates of its extremal points. Note that in $\mathbb{R}^4$, all the 16 children sub-boxes of a given box are adjacent. Our injectivity criterion described in Sect. 7.3 is implemented in a test function (called ***regular***) if it returns false on a sub-box then the sub-box is subdivided.

7.4.2 Subdivision Algorithm

The following Algorithm 4 describes the subdivision method for the topology computation. Some other functions that are needed are described in the sequel.

Algorithm 4: Algorithm for topology in 4D: `topology`$(\mathcal{C}, B, \epsilon)$

```
Input:The curve $\mathcal{C}$, a box $B = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3] \times [a_4, b_4]$ and a tolerance $\epsilon$.
Output:A list of segments in $\mathbb{R}^4$ representing the topology.
Create the hextree $\mathcal{H}$. Initialize the root of $\mathcal{H}$ by $B$ and the intersection points $\mathcal{C} \cap \partial B$.
Create a list of nodes $\mathcal{L}$:  $\mathcal{L} \longleftarrow$ rootOf($\mathcal{H}$).
While $\mathcal{L} \neq \emptyset$, Take the first item $n$ of $\mathcal{L}$ (and remove it from $\mathcal{L}$)
If regular($\mathcal{C}, n$), Then $n \longleftarrow$ regularTopology($\mathcal{C}, n$)
Else If the current box has a size $\geq \epsilon$, Then $\mathcal{L} \longleftarrow$ subdivision ($\mathcal{C}, n$).
Else  Connect all the border points to the center of the box (this applies when we stop the
subdivision because we arrived below the fixed threshold).
Return(fusion($\mathcal{H}$)).
```

Now we describe the other functions called by `topology`.

Function `regular`

This function corresponds to the injectivity criterion described in Sect. 7.3. In fact there are four different tests and each of them corresponds to the fixed variable choice s, t, u or v (see Algorithm 5). If one of them is verified, then we call the corresponding function `regularTopology`.

Function `regularTopology`

If one of the four regularity tests `regular` is verified, then the topology of $\mathcal{C}$ is known. In the function `regularTopology`, we just have to connect the border points in the current node. We also have four different `regularTopology` functions corresponding to the fixed variable s, t, u or v. For example, if s is

Algorithm 5: Injectivity criterion: `regular`($\mathcal{C}, n$)

```
If $\phi$ is locally injective in $n$, Then
  If $\phi_1$ has the wanted monotony on the edges of $n$, Then
    If $\phi_2$ or $\phi_3$ has the wanted monotony on $\partial S_1$,
      Then Return (true), Else Return (false),
        Else If $\phi_2$ has the wanted monotony on the edges of $n$, Then
      If $\phi_1$ or $\phi_3$ has the wanted monotony on $\partial S_2$,
Then Return (true), Else Return (false),
    Else If $\phi_3$ has the wanted monotony on the edges of $n$, Then
    If $\phi_1$ or $\phi_2$ has the wanted monotony on $\partial S_3$, Then Return (true),
  Else Return (false), Else Return (false), Else  Return (false).
```

fixed, then we have, in the current node, a list of even number of border points $p_1, p_2, \ldots, p_{2k-1}, p_{2k}$ (we repeating a point if its multiplicity is even) sorted by their s component. Then, the topology is described by the list of segments: $[p_1, p_2], \ldots, [p_{2k-1}, p_{2k}]$.

Function `subdivision`

This function subdivides the current box creating 16 children as described in Sect. 7.4.1. It allocates the inherited intersection points and computes the new intersection points with the faces of these sub-boxes.

Function `fusion`

This function is called when the construction of the hextree $\mathcal{H}$ is finished. More precisely, each leaf of $\mathcal{H}$ contains the topology in the corresponding sub-box. `fusion` provides the topology of $\mathcal{C}$ in the initial box B. Its implementation consists in merging recursively the topology between the children of each node.

7.4.3 Connected Components and Loops

Algorithm 4 allows to identify the connected components easily. Indeed, the resulted topology of $\mathcal{C}$ is a list of segments $\{[p_1, p_2], \ldots, [p_{2k-1}, p_{2k}]\}$, when k is a positive integer. If there exists $i \in \{1, \ldots, k-1\}$, such that $p_{2i} \neq p_{2i+1}$, then the two segments $[p_{2i-1}, p_{2i}]$ and $[p_{2i+1}, p_{2i+2}]$ are on two different connected components of the topology. A similar simple argument allows to detect the loops (connected) components. The algorithm is illustrated with some examples, that the reader will find at the end of Sect. 7.5.

7.5 Topology in $\mathbb{R}^3$

Sections 7.3 and 7.4 presented an algorithm to compute the topology of a curve $\mathcal{C}$ in $\mathbb{R}^4$ defined by three equations $F(s,t) = G(u,v)$ (with $(s,t,u,v) \in [0,1]^4$ for example). Now, we consider the following projections:

$$\pi_1 : \begin{pmatrix} [0,1]^4 & \longrightarrow \mathbb{R}^2 \\ (s,t,u,v) & \longmapsto (s,t) \end{pmatrix}$$

$$\pi_2 : \begin{pmatrix} [0,1]^4 & \longrightarrow \mathbb{R}^2 \\ (s,t,u,v) & \longmapsto (u,v) \end{pmatrix}.$$

The intersection Γ in $\mathbb{R}^3$ of the two parameterized surface patches $\mathbf{F}$ and $\mathbf{G}$ is the image of $\mathcal{C}$ by $F \circ \pi_1$ (or $G \circ \pi_2$). Our algorithm guarantees (up to the tolerance ϵ) the topology of $\mathcal{C}$, which is isotopic to a collection of segments in $[0,1]^4$. This implies that the image by $F \circ \pi_1$ of a connected component $\mathcal{C}_1$ of $\mathcal{C}$ is connected. However, if $\mathcal{C}_1$ is a loop in $]0,1[^4$ (a closed path) then its image is also a loop in $\mathbb{R}^3$.Moreover, if $\mathcal{C}$ admits several connected components which are loops in $[0,1]^4$, their images by $F \circ \pi_1$ in $\mathbb{R}^3$ may be interlaced (like the Olympic rings). If $\mathcal{C}_1$ is determined by a segment discretization which is too coarse, the knot structure (and the interlacements) can be missed in the image by $F \circ \pi_1$ of this piecewise approximation. We may have the situation depicted in Fig. 7.12.

Similarly, the topology of the projection $\hat{\mathcal{C}}$ of $\mathcal{C} \subset [0,1]^4$ on $[0,1]^2$ by π_1, may not be determined by a coarse discretization of $\mathcal{C}$, even if this discretization is sufficient to determine the topology of $\mathcal{C}$ in $[0,1]^4$, see Fig. 7.13: the self-intersection point is missed. In order to capture these features, the algorithm described in Sects. 7.3 and 7.4 should be extended and the subdivision criteria refined.

As described in Sects. 7.3 and 7.4, we choose a threshold ϵ such that the singular points of the curve Γ will be contained in boxes of a size smaller than ϵ. We aim to determine the topology of the curve Γ up to this indetermination, i.e. two segments entering a box of a size smaller than ϵ are supposed to intersect and form a singular point. All other points are considered smooth.

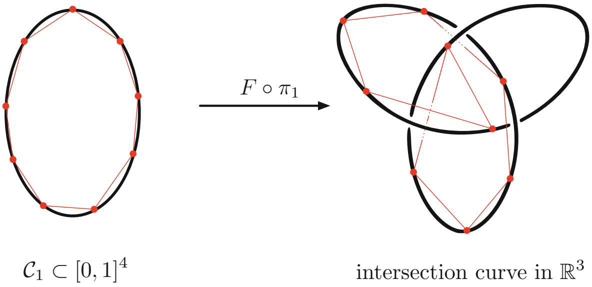

$F \circ \pi_1$
$\mathcal{C}_1 \subset [0,1]^4$
intersection curve in $\mathbb{R}^3$

Fig. 7.12 Image of a loop with knot structure

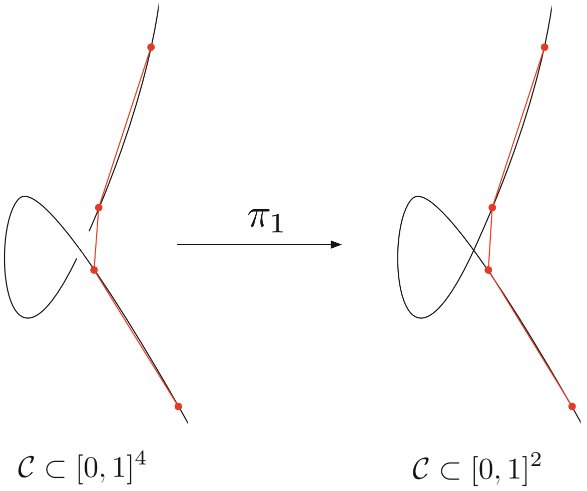


Fig. 7.13 Missing a self-intersection point by projection

Suppose that $\mathcal{C}$ has k loops connected components (when k is a positive integer) denoted respectively by $\mathcal{C}_1, \dots, \mathcal{C}_k$ (we can detect them by using the criterion described in Sect. 7.4.3). We get $\Gamma_i = (F \circ \pi_1)(\mathcal{C}_i)$ for all $i \in \{1, \dots, k\}$.

7.5.1 One Curve Box

Recall that each node of the hextree $\mathcal{H}$ (described in Sect. 7.4.1) stores a box in $\mathbb{R}^4$ and the topology of $\mathcal{C}$ in this box. Let n be a node of $\mathcal{H}$, $\mathcal{B}_n$ be the corresponding box and $B_n = B_F \cap B_G$, where B_F (respectively B_G) is the bounding box constructed with the control points of $F(s,t)$ (respectively $G(u,v)$) written in the Bernstein basis with respect to $\pi_1(\mathcal{B}_n)$ (respectively $\pi_2(\mathcal{B}_n)$). Then, by the convex hull property of the Bézier patches, the bounding box B_n contains the part of Γ corresponding to $\mathcal{B}_n$ i.e. the image of $\mathcal{C} \cap \mathcal{B}_n$ by $F \circ \pi_1$ (or $G \circ \pi_2$).

The discretization of $\mathcal{C}$ is refined, by subdividing all the leaves of $\mathcal{H}$, such that each box (in $\mathbb{R}^4$) intersecting one of the loops $\mathcal{C}_1, \dots, \mathcal{C}_k$ contains at most one segment, i.e. its border intersects $\mathcal{C}$ in two points. Note that in the previous section our algorithm allowed more intersection points. After this step some ambiguities of the node and interlacement structure of Γ may remain. One can see in Fig. 7.14 two bounding boxes (in $\mathbb{R}^3$) sharing interior points. Joining the pairs of points on the borders, the red curve segment may (or may not) pass behind the other green curve segment. So we need to refine the discretization further.

Lemma 7.2 *Let $\gamma_1, \gamma_2 \subset \Gamma$ be two disjoint segments of curves. After a finite number of subdivisions of $(F \circ \pi_1)^{-1}(\gamma_1)$ and $(F \circ \pi_1)^{-1}(\gamma_2)$, the boxes containing γ_1 and the boxes containing γ_2 are disjoint.*

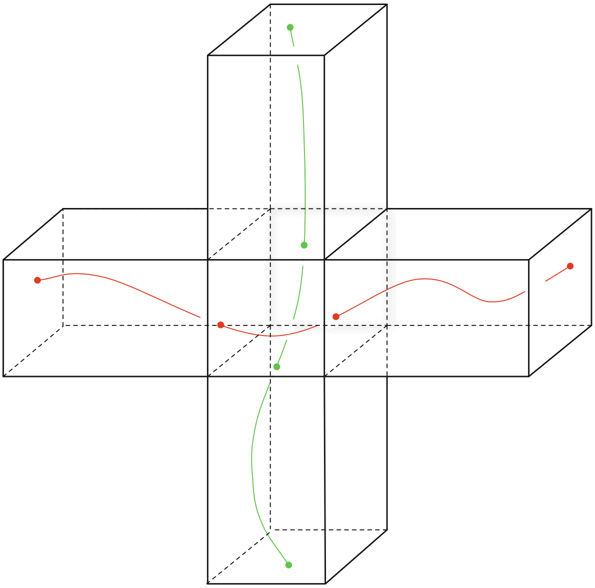


Fig. 7.14 Two boxes sharing interior points

Proof: Indeed by subdivision, the boxes can be made nearer to the curves than the distance between the two curves. ■

The subdivision on the leaves of $\mathcal{H}$ is refined by using Lemma 7.2. Then, we rule out potential ambiguity on interlacements between two loops (situation corresponding the to right picture on Fig. 7.12) because we avoid the situation depicted in Fig. 7.14. The ambiguity on a possible node that is not a loop must still be analyzed.

7.5.2 Node and Discretization

Lemma 7.3 *Let $\gamma \subset \Gamma$ be a segment of curve contained in a bounding box obtained after the subdivision process described in Sect. 7.5.1. Then, the border of this box has just two points p_1 and p_2 of γ. After a finite number of subdivisions of $(F \circ \pi_1)^{-1}(\gamma)$, we have* $\det(N_F, N_G, \overrightarrow{p_1 p_2}) \neq 0$ *(in the corresponding box) with $N_F = \partial_s F \times \partial_t F$ and $N_G = \partial_u G \times \partial_v G$.*

Proof: The condition $\det(N_F, N_G, \overrightarrow{p_1 p_2}) \neq 0$ means that the tangent vector of γ is never orthogonal to $\overrightarrow{p_1 p_2}$. As γ is smooth by hypothesis, the lemma is a consequence of the implicit function theorem. ■

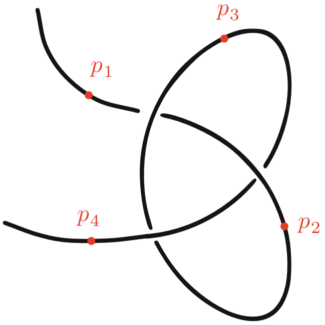


Fig. 7.15 Interlacement situation

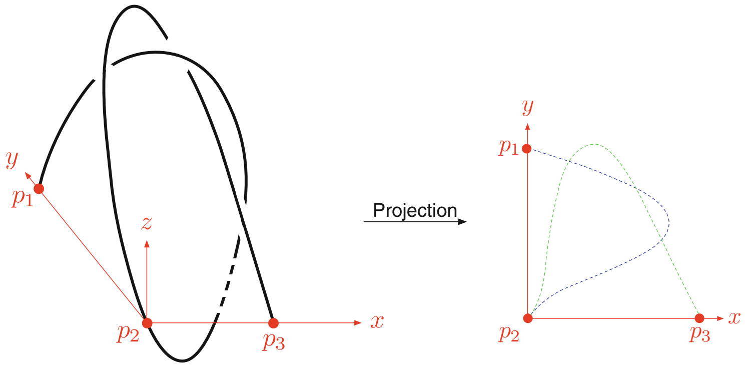


Fig. 7.16 Interlacement with two adjacent segments

If we subdivide the leaves of $\mathcal{H}$ by using Lemma 7.3, then we rule out potential interlacements ambiguities inside each bounding box. However, we still have to avoid interlacing two adjacent branches.

Proposition 7.2 *Let us assume the discretization satisfies Lemmas 7.2 and 7.3 and suppose that* $\det(N_F, N_G, \overrightarrow{p_1p_2}) \neq 0$, $\det(N_F, N_G, \overrightarrow{p_2p_3}) \neq 0$ *and* $\det(N_F, N_G, \overrightarrow{p_1p_3}) \neq 0$ *for two adjacent branches* $[p_1, p_2]$ *and* $[p_2, p_3]$.

If the image (by $F \circ \pi_1$ or $G \circ \pi_2$) of a loop connected component of $\mathcal{C}$ admits a node, then it shows up on the discretization i.e. the sequence of segments obtained by subdivision also describes a node isotopic to that of Γ.

Proof: Indeed, we will have the situation depicted in Fig. 7.15. Since, $\det(N_F, N_G, \overrightarrow{p_1p_2}) \neq 0$, $\det(N_F, N_G, \overrightarrow{p_2p_3}) \neq 0$ and $\det(N_F, N_G, \overrightarrow{p_1p_3}) \neq 0$, we cannot have a node formed by two adjacent segments (depicted in Fig. 7.16). So, we just have to investigate the case when we have at least three segments $[p_1, p_2]$, $[p_2, p_3]$ and $[p_3, p_4]$ (depicted in Fig. 7.15). Lemma 7.3 ensures that each of these segments does not interlace. If the three segments are interlacing, then the bounding boxes containing respectively $[p_1, p_2]$ and $[p_3, p_4]$ intersect each other.■

7.5.3 Examples

We illustrate the algorithm on some examples. First two intersection situations of two polynomial patches are shown on Figs. 7.17 and 7.18. Another classical example (the teapot) is considered. Figure 7.19 shows an approximation of the teapot by 32 biquadratic patches with intersection loci. The resulting topology of these loci is shown on Fig. 7.20.

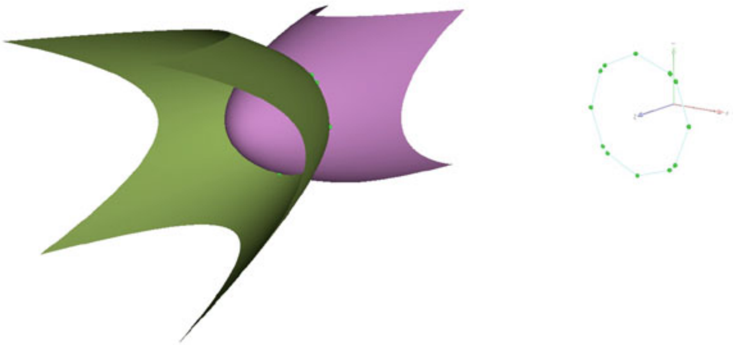


Fig. 7.17 Example of intersection between two polynomial patches

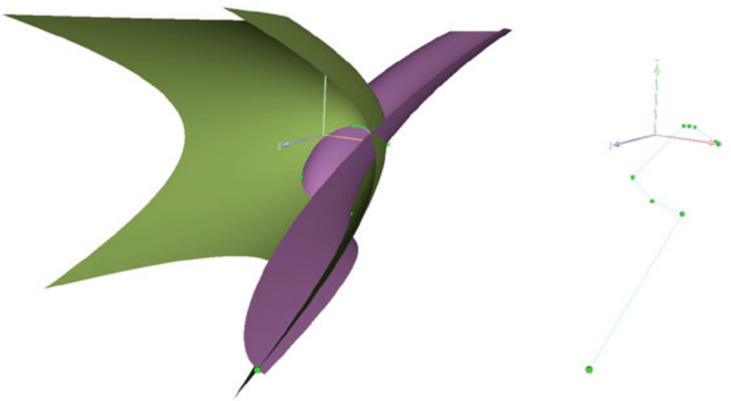


Fig. 7.18 Example of intersection between two polynomial patches

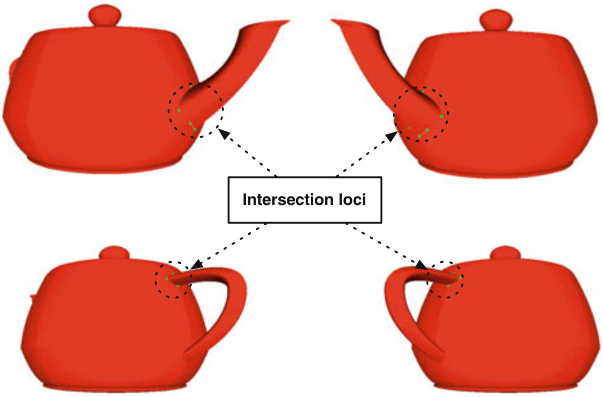


Fig. 7.19 Teapot intersection loci

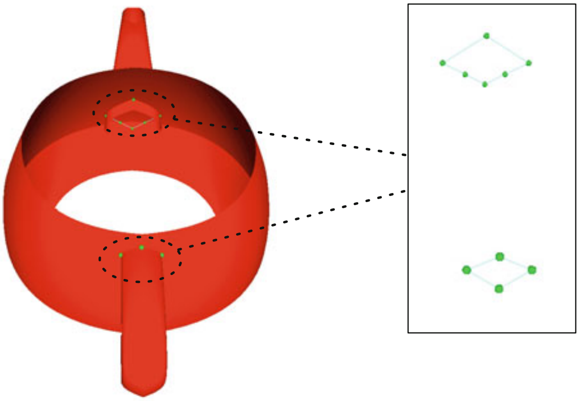


Fig. 7.20 Topology of the teapot intersection

Conclusion

In this paper we have presented a subdivision based algorithm for computing topology of the intersection of two polynomially parameterized surfaces. The computations were performed in a 4D space.

A natural extension would be to treat similarly the intersection of two rationally parameterized surfaces; i.e. introducing a couple of polynomials $F_0(s,t)$ and $G_0(u,v)$ non vanishing on [0, 1] and replacing the initial equations by $\frac{F(s,t)}{F_0(s,t)} - \frac{G(u,v)}{G_0(u,v)}$. All the theoretical developments exposed in this paper rely on differential geometry; they do not explicitly require that the equations are polynomials. So they can be generalized. However the speed of our

(continued)

implementation, which relies on a fast polynomial equations solver, strongly benefited from the fact that we considered polynomials and not rational fractions.

Acknowledgements The authors are grateful to Bernard Mourrain and to the anonymous reviewers for helpful comments and suggestions.

References

1. L. Alberti, B. Mourrain, J. Wintz, Topology and arrangement computation of semi-algebraic planar curves. CAGD **25**(8), 631–651 (2008)
2. J.G. Alcázar, G.M. Diaz-Toca, Topology of 2D and 3D rational curves. Comput. Aided Geom. Des. **27**(7), 483–502 (2010)
3. J.G. Alcázar, J.R. Sendra, Computing the topology of real algebraic space curves. J. Symb. Comput. **39**, 719–744 (2005)
4. D. Arnon, S. McCallum, A polynomial time algorithm for the topological type of a real algebraic curve. J. Symb. Comput. **5**, 213–236 (1988)
5. R.E. Barnhill, S.N. Kersey, A marching method for parametric surface surface intersection. Comput. Aided Des. **7**, 257–280 (1990)
6. M. Barton, G. Elber, I. Hanniel Topologically guaranteed univariate solutions of underconstrained polynomial systems via no-loop and single-component tests. CAD J. **43**(8), 1035–1044 (2011)
7. S. Basu, R. Pollack, M.-F. Roy, *Algorithms in Real Algebraic Geometry.* Volume 10 of Algorithms and Computation in Mathematics, 2nd edn. (Springer, Berlin/New York, 2006)
8. S. Chau, Approximation et intersection des surfaces procédurales utilisées en C.A.O. http://tel.archives-ouvertes.fr/tel-00560289, PhD Thesis, University of Nice, France. (2008)
9. J. Cheng, S. Lazard, L. Penaranda, M. Pouget, F. Rouillier, E. Tsigaridas, On the topology of planar algebraic curves, in *SCG'09 and in Mathematics in Computer Science*, vol. 4 (Aarhus, Denmark, 2014), pp. 113–137
10. J. Cheng, K. Jin, D. Lazard, Certified rational parametric approximation of real algebraic space curves with local generic position method. J. Symb. Comput. **58**, 18–40 (2013)
11. D. Daouda, B. Mourrain, O. Ruatta, On the computation of the topology of a non-reduced implicit space curve, in *Proceedings ISSAC'08*, Hagenberg, pp. 47–54 (2008)
12. T. Dokken, V. Skytt, A.M. Ytrehus, Recursive subdivision and iteration in intersections. Math. Methods Comput. Aided Geom. Des. **5**, 207–214 (1989)
13. M. El Kahoui, Topology of real algebraic space curves. J. Symb. Comput. **43**(4), 235–258 (2008)
14. M. El Kahoui, L. Gonzalez-Vega, An improved upper complexity bound for the topology computation of a real algebraic plane curve. J. Complex. **12**(4), 527–544 (1996)
15. P. Emeliyanenko, A. Kobel, E. Berberich, M. Sagraloff, Exact symbolic-numeric computation of planar algebraic curves. Theor. Comput. Sci. (TCS) **491**, 1–32 (2013)
16. G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 3rd edn. (Academic, Boston, 1993)
17. D. Salomon, D. Bryant, G. Motta, *Handbook of Data Compression*, 5th edn. (Springer, London/New York, 2010)
18. G. Gatellier, A. Labrouzy, B. Mourrain, J.-P. Técourt. *Computing the topology of 3-dimensional algebraic curves.* Computational Methods for Algebraic Spline Surfaces, (Springer, Berlin/Heidelberg, 2005), pp. 27–44.

19. L. González-Vega, I. Necula, Efficient topology determination of implicitly defined algebraic plane curves. Comput. Aided Geom. Des. **19**(9), 719–743 (2002)
20. T.A. Grandine, F.W. Klein, A new approach to the surface intersection problem. Comput. Aided Geom. Des. **14**, 111–134 (1997)
21. J. Hass, R.T. Farouki, C.Y. Han, X. Song, T.W. Sederberg, Guaranteed consistency of surface intersections and trimmed surfaces using a coupled topology resolution and domain decomposition scheme. Adv. Comput. Math. **27**(1), 1–26 (2007)
22. H. Hong, An efficient method for analyzing the topology of plane real algebraic curves. Math. Comput. Simul. **42**, 571–582 (1996)
23. J. Hoschek, D. Lasser, *Fundamental of Computer Aided Geometric Design* (A.K. Peters, Wellesley, 1993)
24. M. Kerber, M. Sagraloff, A worst-case bound for topology computation of algebraic curves. J. Symb. Comput. (JSC) **47.2**, 239–258 (2012). See also: *Improved Complexity Bounds for Computing with Planar Algebraic Curves.*, arxiv.org/1401.5690.
25. C. Liang, B. Mourrain, J.-P. Pavone. *Subdivision methods for the topology of 2d and 3d implicit curves.* Geometric modeling and algebraic geometry, (Springer, Berlin/Heidelberg, 2008), pp. 171–186.
26. S. Plantinga, G. Vegter, Isotopic approximation of implicit curves and surfaces, in *SGP'04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, Nice (ACM, New York, 2004), pp. 245–254
27. T.W. Sederberg, Algorithm for algebraic curve intersection. Comput. Aided Des. **21**(9), 547–554 (1989)
28. T.W. Sederberg, R.J. Meyers, Loops detection in surface patches intersections. Comput. Aided Geom. Des. **5**, 161–171 (1988)

Chapter 8
Rational Bézier Formulas with Quaternion and Clifford Algebra Weights

Rimvydas Krasauskas and Severinas Zubė

8.1 Introduction

Bézier curves and surfaces are widely used in computer graphics and computer-aided design. Their formulas are affine invariant and depend on control points that are visually intuitive and convenient for many applications. On the other hand, there is an important class of primitive surfaces (spheres, rotational cylinders, rotational cones, and tori) with specific properties that are not intrinsic to their classical Bézier representation. For example, one simple reason is the lack of affine invariance.

An alternative theory for curves on a plane was introduced by Sanchez-Reyes in [20]: complex rational Bézier curves were defined using complex numbers for control points and weights. This complex Bézier approach has two main advantages:

- More compact representation: the degree is halved (e.g. circles have linear form);
- Invariance with respect to Möbius transformations.

In order to extend this theory of complex planar curves to surfaces in space we use quaternions and follow the quaternion representation of circles in space [21]. Here one can hardly expect a theory as complete as in the planar case. Indeed, as the quaternion algebra $\mathbb{H}$ is 4-dimensional, one needs to take extra care to ensure that resulting surfaces are contained in the 3-dimensional subspace in $\mathbb{H}$ which is identified with $\mathbb{R}^3$. Also tools for intuitive shape manipulation are still under development.

Of course, there is one notable exception: the case of spherical quaternion curves and surface patches can be reduced to the complex planar case (because a sphere is

R. Krasauskas (✉) • S. Zubė
Vilnius University, Vilnius, Lithuania
e-mail: rimvydas.krasauskas@mif.vu.lt; severinas.zube@mif.vu.lt

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_8

Möbius equivalent to a plane). We only sketch the spherical case in Sect. 8.2.5. and postpone the details to a separate publication.

The first part of the current chapter (Sects. 8.2–8.3) is devoted to the simplest non-trivial case of a quaternion surface, namely a bilinear Quaternion–Bézier (QB) patch. Arbitrary bilinear QB surface patches are characterized as special Darboux cyclide patches using a recent exposition of the classical theory of Darboux cyclides in [19]. Actually, the results of the unpublished manuscript [13] are presented here with extended proofs and more details about Möbius transformations. Note that these new formulas not only reproduce earlier known biquadratic parametrizations of principal patches of Dupin cyclides (considered, e.g. in [2, 11]), but also define totally unknown patches on general Darboux cyclides, which can hardly be generated using the customary Bézier approach.

The second part of the chapter is targeted to a reader who has certain acquaintance with geometric (Clifford) algebras and the isotropic model of Laguerre geometry. Similar Bézier-like formulas make sense in higher dimensional pseudo-Euclidean spaces if quaternions are replaced with elements of the corresponding geometric algebra. Section 8.4 is devoted to general Clifford–Bézier formulas and recent research: the conformal model of Euclidean space, isotropic geometry and isotropic cyclides, and applications to PN-surface modeling.

8.2 Quaternionic Bézier Formulas

8.2.1 Quaternions

We will use the algebra of quaternions $\mathbb{H}$ with the standard basis $\mathbf{1}$, i, j, k:

$$\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = -\mathbf{1}, \quad \mathrm{ij} = \mathrm{k}, \quad \mathrm{jk} = \mathrm{i}, \quad \mathrm{ki} = \mathrm{j}.$$

Reals $\mathbb{R}$ and the 3-dimensional space $\mathbb{R}^3$ will be identified with the *real* and the *imaginary* vector subspaces of $\mathbb{H}$:

$$\mathbb{R} = \mathrm{Re}\,\mathbb{H}, \quad r \mapsto r\mathbf{1}, \qquad \mathbb{R}^3 = \mathrm{Im}\ \mathbb{H}, \quad v \mapsto v_1\mathrm{i} + v_2\mathrm{j} + v_3\mathrm{k}.$$

It will be convenient to decompose a quaternion $q \in \mathbb{H}$ into its scalar (real) part and its vector (imaginary) part:

$$q = r + v = r\mathbf{1} + v_1\mathrm{i} + v_2\mathrm{j} + v_3\mathrm{k}, \quad r = \mathrm{Re}(q), \quad v = \mathrm{Im}\,(q).$$

The quaternionic product in this notation has the following compact formula:

$$qq' = (r + v)(r' + v') = (rr' - v \cdot v') + (rv' + r'v + v \times v'),$$

where $v \cdot v'$ and $v \times v'$ are dot and vector products in $\mathbb{R}^3$.

If $q = r + v$ then $\bar{q} = r - v$ is a conjugate quaternion, $|q| = \sqrt{q\bar{q}}$ is its length, and $q^{-1} = \bar{q}/|q|^2$ is the inverse of q. In particular, if $v \in \mathrm{Im}\ H$ then $\bar{v} = -v$ and $v^{-1} = -v/|v|^2$.

8.2.2 Möbius Transformations in $\mathbb{R}^3$

Möbius (M) transformations in space are generated by inversions in $\mathbb{R}^3$ with respect to spheres. Alternatively, after identifying R^3 with the subset $\mathrm{Im}\,\mathbb{H}$ of imaginary quaternions, M-transformations can be generated by four kinds of elementary transformations: reflections, translations, dilatations, and special inversions (with unit radius and center in the origin)

$$R_v(x) = -vxv, \quad T_a(x) = x + a, \quad S_r(x) = rx, \quad \mathrm{inv}(x) = -x^{-1}, \tag{8.1}$$

where $v, a \in \mathrm{Im}\,\mathbb{H}$, $|v| = 1$, $r \in \mathbb{R}_+$. The composition of an even number of reflections is a rotation.

M-transformations can be represented as fractional-linear functions Φ_A associated with a 2×2 matrix A with quaternion entries:

$$\Phi_A(x) = (ax + b)(cx + d)^{-1}, \quad A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}. \tag{8.2}$$

Usual multiplication of matrices (multiplication of their elements should be in natural order!) corresponds to composition of fractional-linear functions. For example, elementary transformations Eq. (8.1) correspond to the following matrices

$$\begin{pmatrix} v & 0 \\ 0 & v \end{pmatrix}, \quad \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} r & 0 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{8.3}$$

From [3, Theorem 11.1] it follows that the map $A \mapsto \Phi_A$ defines a surjective homomorphism of the matrix group

$$\mathrm{GL}(\mathrm{Im}\,\mathbb{H}) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mid \mathrm{Re}(a\bar{c}) = \mathrm{Re}(b\bar{d}) = 0,\ \bar{b}c + \bar{d}a \in \mathbb{R}^* \right\}, \quad \mathbb{R}^* = \mathbb{R} \setminus \{0\},$$

to the group of M-transformations of $\mathrm{Im}\,\mathbb{H}$.

For any for points $p_0, \ldots, p_3 \in \mathrm{Im}\,\mathbb{H}$ we define a *cross-ratio*

$$\mathrm{cr}(p_0, p_1, p_2, p_3) = (p_0 - p_1)(p_1 - p_2)^{-1}(p_2 - p_3)(p_3 - p_0)^{-1}. \tag{8.4}$$

There is a full analog of the well-known fact that four complex points lie on a circle only if their cross-ratio is real.

Proposition 8.1 **([4])** *A cross-ratio* $\mathrm{cr}(p_0, p_1, p_2, p_3)$ *is real if and only if these four points are on a circle. A real cross-ratio is Möbius invariant.*

8.2.3 Properties of Quaternionic Bézier Formulas

A *quaternionic Bézier (QB) formula* is a fraction of two linear combinations of control points and weights $p_i, w_i \in \mathbb{H}$ with coefficients B_i

$$F = \left(\sum_i p_i w_i B_i\right)\left(\sum_i w_i B_i\right)^{-1}.$$

The coefficients B_i can be the following Bernstein polynomials:

- $B_i^d(t) = \binom{d}{i}(1-t)^{d-i}t^i$, $i = 0, \ldots, d$ define *QB curves* of degree d;
- $B_i^{d_1}(s)B_j^{d_2}(t)$, $i = 0, \ldots, d_1$, $j = 0, \ldots, d_2$, define *QB tensor product surfaces* of bidegree (d_1, d_2);
- $B_{ij}^d = \frac{d!}{(d-i-j)!i!j!}(1-s-t)^{d-i-j}s^i t^j$, with integers $i \geq 0$, $j \geq 0$, $i + j \leq d$, define *QB triangular surfaces* of degree d.

General QB formulas take values in the 4-dimensional space of quaternions $\mathbb{H}$. The most interesting space for us will be $\mathbb{R}^3$, which is identified with $\mathrm{Im}\,\mathbb{H}$. Therefore, we always need to ensure that the QB curves and surfaces under consideration are contained in $\mathrm{Im}\,\mathbb{H}$.

Proposition 8.2 *A QB formula* $F = (\sum_i p_i w_i B_i)(\sum_i w_i B_i)^{-1}$ *is invariant with respect to Möbius transformations: if* $\Phi = \Phi_A$ *defined by* (8.2) *then*

$$\Phi(F) = (\textstyle\sum_i p_i' w_i' B_i)(\sum_i w_i' B_i)^{-1}, \quad p_i' = \Phi(p_i),\ w_i' = \hat{\Phi}(p_i, w_i) = (cp_i + d)w_i.$$

If $F \in \mathrm{Im}\,\mathbb{H}$ *then* $\Phi(F) \in \mathrm{Im}\,\mathbb{H}$.

Proof It is easy to check this directly using the identities

$$\Phi(xy^{-1}) = (ax + by)(cx + dy)^{-1}, \quad p_i' w_i' = \Phi(p_i)\Phi(p_i, w_i) = (ap_i + b)w_i.$$

□

This proposition allows us to calculate Möbius transformations of any rational Bézier curves and surfaces (with real weights) and get a lot of new non-trivial examples of QB curves and surfaces in $\mathrm{Im}\,\mathbb{H}$.

Remark 8.1 Möbius deformations of 3D models were realized on GPU using real time evaluations of quaternion formulas in [9] (see Fig. 8.1). Since such transformations are conformal, they may be convenient for deforming organic shapes including textures.

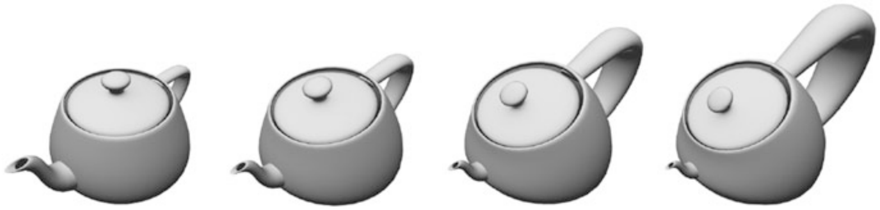


Fig. 8.1 Möbius transformation of the Utah Teapot [9]

In general the quaternion representation has half the degree of the customary rational Bézier case with real weights.

Proposition 8.3 *A QB formula* $F = (\sum_i p_i w_i B_i)(\sum_i w_i B_i)^{-1}$ *of degree* d *(resp. bidegree* (d_1, d_2)*) defines the same parametrization as the following Bézier formula of degree* $2d$ *(resp. bidegree* $(2d_1, 2d_2)$*)*

$$\tilde{F} = (N\bar{D})(D\bar{D})^{-1}, \quad N = \sum_i p_i w_i B_i, \quad D = \sum_i p_i w_i B_i.$$

Proof Note that the new numerator $N\bar{D}$ and denominator $D\bar{D}$ can both be expanded in the new basis $\{B_i B_j\}$ and $D\bar{D}$ is real. □

8.2.4 Circular Arcs

Here we reformulate some results of [21] about the quaternionic representation of a circle and give proofs based on Propositions 8.2 and 8.3.

Let p_0 and p_1 be the two endpoints of a circular arc C in $\mathbb{R}^3 = \operatorname{Im}\mathbb{H}$, and let p_∞ be some point on the complementary arc of C. We are going to parametrize this arc rationally in three steps:

- Apply the inversion $I : x \mapsto p_\infty - (x - p_\infty)^{-1}$ with the center in p_∞ to the circle C;
- Parametrize the resulting line segment $L(t) = I(p_0)(1-t) + I(p_1)t$;
- Apply the same inversion once more $C(t) = I(L(t))$.

The inversion I is a composition $T_{p_\infty} \circ \operatorname{inv} \circ T_{-p_\infty}$ of elementary transformations (8.1) and has the following matrix representation (see (8.3))

$$\begin{pmatrix} 1 & p_\infty \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & -p_\infty \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} p_\infty & -1 - p_\infty^2 \\ 1 & -p_\infty \end{pmatrix}.$$

According to Proposition 8.2, a linear Bézier curve $L(t)$ with control points $I(p_0)$, $I(p_1)$ and unit weights is transformed to a linear QB curve with control points

$I(I(p_i)) = p_i$, and weights $w_i = I(p_i) - p_\infty = -(p_i - p_\infty)^{-1}$, $i = 0, 1$. So we finally get the parametrization of the circular arc

$$C(t) = (p_0 w_0(1-t) + p_1 w_1 t)(w_0(1-t) + w_1 t)^{-1}, \tag{8.5}$$

that is contained in Im $\mathbb{H}$ by construction.

Remark 8.2 One can divide both weights in (8.5) by w_0 and get the equivalent pair of weights $w'_0 = 1$ and $w'_1 = w_1 w_0^{-1}$. If just one weight is multiplied by a real number $\lambda > 0$ then the arc is reparametrized.

Remark 8.3 The parameter t in $C(t)$ has a simple interpretation as a cross-ratio

$$t = \mathrm{cr}(p_\infty, p_1, p_0, C(t)). \tag{8.6}$$

Indeed, according to Proposition 8.1 it is enough to check the following much simpler identity $\mathrm{cr}(\infty, I(p_1), I(p_0), L(t)) = t$, where $\infty = I(p_\infty)$ is the infinite point of Im $\mathbb{H}$.

The customary Bézier representation of $C(t)$ can be derived using Proposition 8.3: denote the numerator of the fraction (8.5) by N and the denominator by D, then rewrite this fraction as $C(t) = ND^{-1} = N\bar{D}(D\bar{D})^{-1}$ with a real denominator, and expand both $N\bar{D}$ and $D\bar{D}$ in the quadratic Bernstein basis B_i^2, $i = 0, 1, 2$:

$$N\bar{D} = p_0 w_0 \bar{w}_0 B_0^2 + \frac{1}{2}(p_0 w_0 \bar{w}_1 + p_1 w_1 \bar{w}_0) B_1^2 + p_2 w_2 \bar{w}_2 B_2^2,$$
$$D\bar{D} = w_0 \bar{w}_0 B_0^2 + \frac{1}{2}(w_0 \bar{w}_1 + w_1 \bar{w}_0) B_1^2 + w_2 \bar{w}_2 B_2^2.$$

Hence $C(t)$, as a quadratic rational Bézier curve, has real weights

$$W_0 = w_0 \bar{w}_0, \quad W_1 = \frac{1}{2}(w_0 \bar{w}_1 + w_1 \bar{w}_0) = \mathrm{Re}(w_0 \bar{w}_1), \quad W_2 = w_1 \bar{w}_1, \tag{8.7}$$

and control points

$$P_0 = p_0, \quad P_1 = \frac{p_0 w_0 \bar{w}_1 + p_1 w_1 \bar{w}_0}{w_0 \bar{w}_1 + w_1 \bar{w}_0}, \quad P_2 = p_1. \tag{8.8}$$

We can also calculate a tangent vector v_0 to $C(t)$ at p_0 as a derivative $C'(0)$ at $t = 0$. First we differentiate the identity $N = CD$ and get $N' = C'D + CD'$. Then $C'(0) = (N'(0) - C(0)D'(0))D(0)^{-1} = (p_1 w_1 - p_0 w_0 - p_0(w_1 - w_0))w_0^{-1}$ and

$$v_0 = C'(0) = (p_1 - p_0) w_1 w_0^{-1}. \tag{8.9}$$

Therefore, the weights cannot be arbitrary. The following relations between weights and control points will be useful later.

Proposition 8.4 *If C is a QB curve* (8.5) *with $p_0, p_1 \in \operatorname{Im}\mathbb{H}$ then:*

$$C \subset \operatorname{Im}\mathbb{H} \Leftrightarrow \operatorname{Re}(p_0 w_0 \bar{w}_1 + p_1 w_1 \bar{w}_0) = 0 \tag{8.10}$$

$$C \subset \operatorname{Im}\mathbb{H} \Leftrightarrow \operatorname{Im}(w_1 w_0^{-1}) \perp (p_1 - p_0) \tag{8.11}$$

$$C \text{ is a line} \Leftrightarrow w_1 w_0^{-1} \in \mathbb{R} \tag{8.12}$$

If $C \subset \operatorname{Im}\mathbb{H}$ and $u = \operatorname{Im}(w_1 w_0^{-1}) \neq 0$ then C is a circle in a plane orthogonal to u.

Proof The circle C is contained in $\operatorname{Im}\mathbb{H}$ if and only if its middle control point P_1 is there, i.e. $\operatorname{Re} P_1 = 0$. From its expression in (8.8) condition (8.10) follows. This is also equivalent to $v_0 \in \operatorname{Im}\mathbb{H}$, so $(p_1 - p_0)w_1 w_0^{-1} \in \operatorname{Im}\mathbb{H}$ (see (8.9)) and (8.11) follows. Similarly (8.12) and the last statement can be derived from (8.9). □

8.2.5 Spherical Quaternionic Bézier Curves and Surface Patches

In this section we sketch the theory of QB curves and surfaces on a sphere (or plane), by reducing quaternionic-Bézier formulas to complex-Bézier formulas.

Let us start with the simplest surface example. A spherical triangle with corner points p_0, p_1, p_2 bounded by three circular arcs (such that these three circles intersect in a point p_∞) has the parametrization formula with weights: $w_i = -(p_i - p_\infty)^{-1}$, $i = 0, 1, 2$, (see Fig. 8.2, left)

$$T(s,t) = (p_0 w_0 (1-s-t) + p_1 w_1 s + p_2 w_2 t)(w_0(1-s-t) + w_1 s + w_2 t)^{-1}. \tag{8.13}$$

Indeed this is the Möbius image of a planar linear Bézier triangle (cf. Sect. 8.2.4).

In order to generalize this example we introduce two different inclusions of $\mathbb{C}$ into $\mathbb{H}$ (which are compatible with geometric algebra formulas in Sect. 8.4.3)

$$\mathrm{in}_1 : \mathbb{C} \to \mathbb{H},\ x + y\mathrm{i} \mapsto x - y\mathrm{k}, \quad \mathrm{in}_2 : \mathbb{C} \to \operatorname{Im}\mathbb{H},\ x + y\mathrm{i} \mapsto x\mathrm{i} + y\mathrm{j}. \tag{8.14}$$

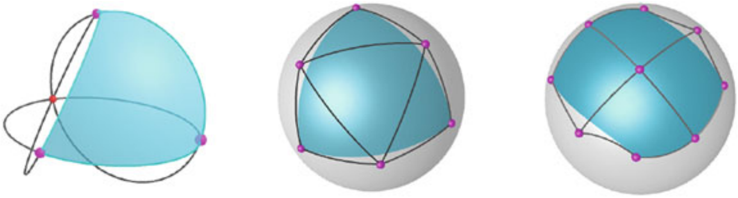


Fig. 8.2 Spherical QB-patches: linear and quadratic triangle, biquadratic quad patch

Then any complex-Bézier formula $(\sum_i p_i w_i B_i)(\sum_i w_i B_i)^{-1}$, $p_i, w_i \in \mathbb{C}$, can be transformed to a QB formula $(\sum_i p_i' w_i' B_i)(\sum_i w_i' B_i)^{-1}$, $w_i' = \mathrm{in}_1(w_i)$, $p_i' = \mathrm{in}_2(p_i)$, on the plane $z = 0$ or on any sphere in $\mathrm{Im}\,\mathbb{H}$ by Möbius invariance according to Proposition 8.2. Hence the whole theory of complex Bézier curves on a plane developed in [20] can be translated into QB form and then extended to any sphere.

Even more, it appears that all rational Bézier curves and surface patches on a sphere can be represented in QB form of halved degree. The following theorem can be proved using generalized stereographic projection [6] and its interpretation in terms of complex projective line [10].

Theorem 8.1 *Any rational Bézier curve or triangular (resp. rectangular) surface patch of degree* $2d$ *(resp. of bidegree* $(2d_1, 2d_2)$*) on a sphere* $S^2 \subset \mathrm{Im}\,\mathbb{H}$ *can be represented in a quaternionic Bézier form of degree* d *(resp. of bidegree* (d_1, d_2)*) with the control net composed of circular arcs lying on* S^2*.*

Figure 8.2 (middle and right) illustrates a QB-triangle of degree 2 (which is a spherical octant) and a biquadratic QB-rectangle.

The quaternionic approach allows us to deal with all spheres in $\mathbb{R}^3$ in the unique framework. In a certain sense every sphere carries its own complex structure that is encoded in global quaternionic structure.

8.3 Bilinear Quaternionic Bézier Patches

We do not know much about general QB curves and surfaces, so we are going to study important particular cases.

Remark 8.4 Non-spherical QB-curves of degree 2 in $\mathrm{Im}\,\mathbb{H}$ are characterized in [22] as the diagonals $P(t, t)$ of bilinear QB-surfaces defined by (8.15). Note that the middle control point of such curves is not contained in $\mathrm{Im}\,\mathbb{H}$.

The simplest cases of QB surfaces are linear triangles and bilinear quadrangles. The first case will turn out to be spherical, hence we will focus on the latter.

Proposition 8.5 *Any linear QB triangle in* $\mathrm{Im}\,\mathbb{H}$ *is spherical.*

Proof Consider the formula of a QB triangular patch (8.13), and apply inversion with center on the boundary circle going through p_0 and p_1 that transforms this circle into a line. Using the same notation for control points and weights, one can assume $w_0 = w_1 = 1$ (see Remark 8.2 and (8.12)). Then it follows from Proposition 8.4 that $\mathrm{Im}\,(w_2)$ is orthogonal to a family of circles with control points $q(s) = p_0(1 - s) + p_1 s$, p_2 and weights $1, w_2$, that cover the patch. Hence, this inversion of the initial triangular patch is planar. □

8.3.1 Properties of Bilinear QB-Patches

Let us define a bilinear QB-quadrangular patch (call it just a bilinear QB-patch) with slightly different indexing (the fraction is used in the sense $\frac{a}{b} = ab^{-1}$):

$$P(s,t) = \frac{p_0w_0(1-s)(1-t) + p_1w_1s(1-t) + p_2w_2(1-s)t + p_3w_3st}{w_0(1-s)(1-t) + w_1s(1-t) + w_2(1-s)t + w_3st}, \tag{8.15}$$

We consider only the case when the image is contained in Im $\mathbb{H} = \mathbb{R}^3$.

Lemma 8.1 *If two adjacent boundary circles of a bilinear QB-patch P defined by (8.15) are cospherical then the patch is either spherical or a patch of a double ruled quadrics (including its Möbius transformations).*

Proof Denote by C_{ij}, $ij = 01, 02, 13, 23$, the boundary circles connecting adjacent points p_i and p_j. If circles C_{01} and C_{02} are cospherical then there are two cases: (a) they intersect in two points p_0 and $q \neq p_0$; (b) they are tangent in p_0 (a double point).

In case (a) we apply inversion with a center q and use the same notation for the transformed patch. Now C_{01} and C_{02} are line segments, and one can assume (after a reparametrization) $w_0 = w_1 = w_2 = 1$ (see (8.12)). If $w_3 \in \mathbb{R}$ then the patch P is on a bilinear quadric (or plane). Otherwise Im $(w_3) \neq 0$, and according to Proposition 8.4 Im $(w_3) \perp (p_3 - p_1)$ and Im $(w_3) \perp (p_3 - p_2)$. Hence the two boundary circles C_{13} and C_{23} lie on the same plane Π going through three points p_1, p_2, p_3. All the weights along these circles have the same direction, since they are linear averages between w_3 and 1. Similarly it follows that all other circles on P are on the same plane Π, and P is planar, i.e. spherical. Case (b) can be treated similarly: apply an inversion with center p_0 and notice that despite the blown-up corner p_0 the same arguments are valid. □

Lemma 8.2 *Let four circles C_{ij}, $ij = 01, 02, 13, 23$, in* Im $\mathbb{H}$ *be defined by pairs of control points and weights $\{(p_i, w_i), (p_j, w_j)\}$, and suppose that any two adjacent circles are not cospherical. Then there is a unique non-zero number*

$$\lambda = -\mathrm{Re}(p_1w_1\bar{w}_2 + p_2w_2\bar{w}_1)\big(\mathrm{Re}(p_0w_0\bar{w}_3 + p_3w_3\bar{w}_0)\big)^{-1} \in \mathbb{R}, \tag{8.16}$$

such that the same control points with weights w_0, w_1, w_2, λw_3 define a bilinear QB-patch in Im $\mathbb{H} = \mathbb{R}^3$.

Proof Denoting numerator and denominator in formula (8.15) with control points $p_i, i = 0, \ldots, 3$, and weights w_0, w_1, w_2, λw_3 by N and D, we can modify it to the form with a real denominator $P = ND^{-1} = N\overline{D}(D\overline{D})^{-1}$. Then we expand $N\overline{D}$ in a biquadratic Bernstein basis and get control points (multiplied by their weights) of the corresponding rational biquadratic Bézier surface. Boundary control points are obviously in $\operatorname{Im}\mathbb{H}$, since they represent circular arcs in $\operatorname{Im}\mathbb{H}$. The middle control point multiplied by its weight has the following form:

$$q_{11} = (p_1 w_1 \bar{w}_2 + p_2 w_2 \bar{w}_1) + \lambda(p_0 w_0 \bar{w}_3 + p_3 w_3 \bar{w}_0),$$

where both expressions in brackets have non-zero real parts (otherwise, adjacent boundary circles will be cospherical, cf. the proof of Proposition 8.5 and formula (8.10)). Solving the equation $\operatorname{Re}(q_{11}) = 0$ for λ we get exactly (8.16). □

Lemma 8.3 *Let C_{02}, C_{01}, C_{13} be circles in $\mathbb{R}^3$, and suppose that $p_0 = C_{02} \cap C_{01}$ and $p_1 = C_{01} \cap C_{13}$ are unique points of their transversal intersection. Then for any other point $p_2 \in C_{02}$, $p_2 \neq p_0$, there exists a unique bilinear QB-patch (up to trivial reparametrization) with control points p_i, $i = 0, 1, 2$, and $p_3 \in C_{13}$ with three boundary arcs lying on the given three circles.*

Proof Our goal is to construct a closed contour of circular quaternionic arcs and then fill the contour using Lemma 8.2.

We choose any point $q \in C_{13}$, $q \neq p_1$, and apply inversion with center q. Using the same notation, we see that C_{13} is a line, and we can find unique (up to real multiplier) weights $w_1 = 1$, w_0 and w_2, that allows us to parametrize the circles C_{01}, C_{02}. The point p_2 and weight w_2 determine a plane Π where a circle C_{23} should be (see Proposition 8.4). So we can find a point p_3 as an intersection $\Pi \cap C_{13}$ with a weight $w_3 = 1$. An exceptional case when Π is parallel to the line C_{13} can happen only when the initial point q (before inversion) can be chosen as p_3. □

8.3.2 Implicitization

We are going to find the implicit equation of the patch (8.15) as an algebraic surface in $\mathbb{R}^3$.

Let us consider a formal equation with a quaternion $X = u + x\mathrm{i} + y\mathrm{j} + z\mathrm{k}$ on the left side and with a bilinear quaternionic patch on the right side:

$$X = N(s,t)D(s,t)^{-1}, \tag{8.17}$$

where $N(s,t)$ and $D(s,t)$ are the numerator and the denominator of the fraction in (8.15). Let us multiply both sides of (8.17) by $D(s,t)$ and move all terms to the left side

$$XD(s,t) - N(s,t) = 0.$$

We treat this quaternionic equation as a system of 4 real linear equations with 4 unknowns

$$(1-s)(1-t), \quad s(1-t), \quad (1-s)t, \quad st.$$

The 4×4 matrix M of this system has 4 columns filled with components of quaternions $(X - p_i)w_i$, $i = 0, \ldots, 3$. Hence the entries of the matrix M are linear forms in u, x, y, z, and the polynomial

$$F(u, x, y, z) = \det\left[(X - p_0)w_0, (X - p_1)w_1, (X - p_2)w_2, (X - p_3)w_3\right] \tag{8.18}$$

must vanish on every point X of the patch $P(s, t)$. Therefore, $F(u, x, y, z) = 0$ defines at most a quartic equation in the variables u, x, y, z.

Theorem 8.2 *Let $P(s, t)$ be the bilinear parametrization of the patch 8.15 in* $\operatorname{Im} \mathbb{H} = \mathbb{R}^3$. *Then an implicit equation of the corresponding parameterized surface is a factor of the polynomial $F(0, x, y, z)$ defined by (8.18) and has at most degree* 4.

Example 8.1 A bilinear QB-patch $P(s, t)$ with the following points $[p_0, \ldots, p_3] = [-\mathrm{i}, \mathrm{i}, -\mathrm{i}+\mathrm{j}, \mathrm{i}+\mathrm{j}]$ and weights $[w_0, \ldots, w_3] = [1, \mathrm{j}, 1, \mathrm{j}]$ generates the equation of the cylinder $x^2 + z^2 - 1 = 0$. The same points as above with the weights $(w_0, \ldots, w_3) = (1, \mathrm{j}, \mathrm{k}, \mathrm{i})$ generate the equation of the torus

$$\left(x^2 + \left(y - \frac{1}{2}\right)^2 + z^2 + \frac{3}{4}\right)^2 - 4\,(x^2 + z^2) = 0.$$

All these examples can be classified as Darboux cyclide patches. *Darboux cyclides* are quartic surfaces with a double conic $x^2 + y^2 + z^2 = 0$ at infinity and their Möbius transformations: non-spherical quadrics and cubics with the same double conic. We consider only irreducible cases: for example, a union of two spheres is excluded.

Corollary 8.1 *Any non-spherical bilinear QB-patch is a Darboux cyclide patch.*

Proof According to Theorem 8.2 a bilinear QB-patch has at most degree 4. Since it is Möbius invariant, its arbitrary inversion is also a bilinear QB-patch. These are sufficient conditions for the patch to be on a Darboux cyclide (see details in [15]).

□

8.3.3 Bilinear Quaternionic Bézier Patches on Darboux Cyclides

It is known from the theory of Darboux cyclides (see the exposition in [19]) that they can contain at most 6 real families of circles, that are grouped in pairs. Two circles from distinct families intersect in a unique point if these families are from different pairs, otherwise the circles are cospherical [19, Propositions 4, 5]. Any bilinear QB-patch defines a Darboux cyclide by Corollary 8.1 and generates two families of isoparametric circles on it.

Theorem 8.3 *Any two families of circles from different pairs on a given Darboux cyclide are generated by a bilinear QB-patch. Two families of circles from the same*

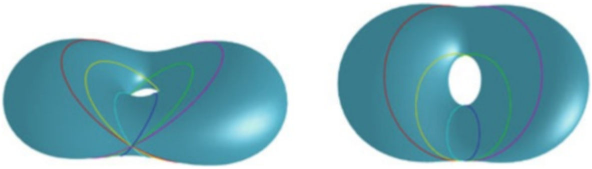


Fig. 8.3 A Darboux cyclide with six circles representing six distinct families

pair can be generated only by a QB-patch defined by rulings of a double ruled quadric (or its Möbius equivalent).

Proof Take two circles C_{02}, C_{13} from one family and one C_{01} from another family that is not in the same pair. Then pairs of circles C_{02}, C_{01} and C_{01}, C_{13} intersect in the unique points p_0 and p_1. Hence we are in the situation of Lemma 8.3 that allows us to construct a bilinear QB-patch bounded by these three circles. So it is enough to prove the uniqueness of a Darboux cyclide going through these circles. Here we can follow [19] and employ the conformal model by lifting the whole construction to a 3-sphere S^3 in the space $\mathbb{R}^4$. Now the circles C_{02}, C_{13} are contained in two 2-planes which intersect in the apex of the quadratic 3-dimensional cone, which cuts our Darboux cyclide in S^3. Let us cut the cone by any hyperplane Π (not containing the apex) and project all circles from the apex to Π. Their images will be two skew lines L_{02}, L_{13} and a conic C'_{01} intersecting them. Therefore the uniqueness problem is reduced to the following simple one: prove the uniqueness of a quadric surface in $\mathbb{R}^3$ going through a given pair of skew lines and one conic. The second part of the theorem follows from Lemma 8.1, since circles from the paired families are cospherical. □

In Fig. 8.3 below we can see an example of a symmetric Darboux cyclide with six paired families of circles (i.e. there are three pairs).

Corollary 8.2 *There are exactly 12 different bilinear QB-patches on a Darboux cyclide with 6 real families of circles.*

Proof Apply Theorem 8.3 and count cases: three choices of two pairs of circle families times four choices of two families from these two distinct pairs. □

8.3.4 *Principal Dupin Cyclide Patches*

For a definition of a Dupin cyclide see e.g. [5] and references therein. This is a particular case of a Darboux cyclide containing two self-paired families of circles (i.e. both families in a pair coincide, see previous Sect. 8.3.3). A *principle Dupin cyclide patch* is a quadrangular patch bounded by circles from these families (see Fig.8.3), which can be characterized by the following properties:

- All angles are right angles and corner points $p_0, \ldots, p_3$ are on a circle;

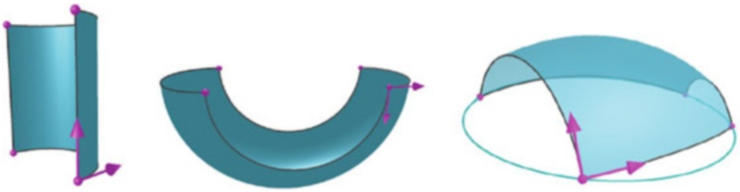


Fig. 8.4 Principal patches on: cilinder, torus, general Dupin cyclide

- Tangent vectors at the end points of the opposite boundary arcs in the adjacent corners are symmetric with respect to a vector joining these points; e.g. let v_{ij} be a tangent vector at p_i pointing to p_j then $v_{23} = -(p_2 - p_0)v_{01}(p_2 - p_0)^{-1}$.

Theorem 8.4 *A principal Dupin cyclide patch with corners in four points $p_0, \dots, p_3$ on a circle and two orthogonal tangent vectors v_{01} and v_{02} at p_0 can be parametrized by a bilinear QB-patch with these control points and the following weights (where $q_{ij} = (p_i - p_j)/|p_i - p_j|$):*

$$w_0 = 1, \quad w_1 = q_{01}v_{01}, \quad w_2 = q_{02}v_{02}, \quad w_3 = |p_2 - p_1||p_3 - p_0|^{-1}q_{13}w_1q_{20}w_2.$$

The proof of this theorem and other important quaternionic formulas related to principal Dupin cyclide patches can be found in [14] (Fig. 8.4).

8.4 Clifford–Bézier Formulas

In this section we collect several extensions of the quaternionic approach, showing that they can be unified in the framework of geometric algebra (associated with the most general case of pseudo-Euclidean spaces). Sections 8.4.1–8.4.4 require basic knowledge of geometric algebra (e.g. [7, 8, 16]). Preliminaries for Sect. 8.4.5 include elements of Laguerre geometry and Pythagorean-normal surfaces [12, 17, 18].

8.4.1 Pseudo-Euclidean Space and Its Geometric Algebra

A *pseudo-Euclidean* space is a vector space $\mathbb{R}^n_\sigma$, with a scalar (interior) product having signature $\sigma = (n_+, n_-, n_0)$, $n = n_+ + n_- + n_0$, i.e. with an orthonormal basis $\{e_1, \dots, e_n\}$, such that

$$e_i \cdot e_i = \begin{cases} 1, & \text{if } i \le n_+, \\ -1, & \text{if } n_+ + 1 \le i \le n_+ + n_-, \\ 0, & \text{if } i > n_+ + n_-, \end{cases} \qquad e_i \cdot e_j = 0,\ i \ne j. \tag{8.19}$$

An abbreviated notation for signature will be used $\sigma = (n) = (n, 0, 0)$ and $\sigma = (n_+, n_-) = (n_+, n_-, 0)$.

Define a *geometric algebra* $\mathcal{G}_\sigma = \mathcal{G}(\mathbb{R}^n_\sigma)$ as a Clifford algebra generated by the pseudo-Euclidean space $\mathbb{R}^n_\sigma$ with a signature $\sigma = (n_+, n_-, n_0)$. The *geometric* product is defined to be associative and distributive with respect to addition, with the extra relation for vectors $v \cdot v = v^2 \in \mathbb{R}$. If $v, u \in \mathbb{R}^n_\sigma$ then the geometric product is a sum of interior and exterior (see below) products

$$vu = v \cdot u + v \wedge u. \tag{8.20}$$

The algebra $\mathcal{G}_\sigma$ has the same underlying vector space as the usual exterior algebra $\bigwedge(\mathbb{R}^n_\sigma)$, namely it is a vector space of dimension 2^n, that can be decomposed as a direct sum $E_0 \oplus E_1 \oplus \cdots \oplus E_n$ of subspaces with the following bases

$$\{1\}, \quad \{e_1, \ldots, e_n\}, \quad \{e_{ij} \mid i < j\}, \quad \{e_{ijk} \mid i < j < k\}, \ldots \quad \{I = e_{12\ldots n}\}, \tag{8.21}$$

where $e_{ij\ldots k} = e_i e_j \cdots e_k$. The vector spaces $E_0, \ldots, E_n$ are scalars, vectors, bivectors, etc. respectively. The basis of E_n has only one element $I = e_1 e_2 \cdots e_n$, which is called a *pseudoscalar*. A *dual* of $x \in \mathcal{G}_\sigma$ is $x^* = Ix$. For any $x \in \mathcal{G}_\sigma$, its k-grade component $\langle x \rangle_k$ is the projection to the subspace E_k of grade k.

A *reversion* operation in the algebra $\mathcal{G}_\sigma$ is defined as follows (see [7, 8, 16] for details). If x is a product of vectors $x = v_1 v_2 \cdots v_{n-1} v_n$, then its reversion is $\tilde{x} = v_n v_{n-1} \cdots v_2 v_1$. If all v_i are non-zero, then $x\tilde{x} = (v_n \cdot v_n) \cdots (v_2 \cdot v_2)(v_1 \cdot v_1) \in \mathbb{R}$. Hence it is easy to calculate the inverse element $x^{-1} = \tilde{x}/(x\tilde{x})$.

8.4.2 *Möbius Transformations in* $\mathbb{R}^n_\sigma$

A group of *Möbius transformations* Möb($\mathbb{R}^n_\sigma$) of a pseudo-Euclidean space $\mathbb{R}^n_\sigma$ is generated by: pseudo-Euclidean reflections, translations, dilatations, and special inversions

$$R_v(x) = -vxv, \quad T_a(x) = x + a, \quad S_r(x) = rx, \quad \mathrm{inv}(x) = x^{-1}, \tag{8.22}$$

where $v, a \in \mathbb{R}^n_\sigma$, $|v| = 1$, $r \in \mathbb{R}_+$. Note the different sign in the inversion formula compared with the quaternionic case (8.1).

Similar to Sect. 8.2.2 M-transformations of $\mathbb{R}^n_\sigma$ can be represented by 2×2 matrices A (with entries in $\mathcal{G}_\sigma$) and corresponding fractional-linear functions Φ_A, see (8.2).

We define *Clifford–Bézier surfaces* (CB-surfaces) by the same rational Bézier formulas treating them as formulas in $\mathcal{G}_\sigma$, i.e. with control points $p_{ij} \in \mathbb{R}^n_\sigma$ and weights $w_{ij} \in \mathcal{G}_\sigma$.

Proposition 8.6 *CB formulas* $(\sum_i p_i w_i B_i)(\sum_i w_i B_i)^{-1}$ *with control points* $p_i \in \mathbb{R}^n_\sigma$ *and weights* $w_i \in \mathcal{G}_\sigma$ *(with certain kind of Bernstein polynomials* B_i*) are mapped by an M-transformation* $\Phi(x) = (ax+b)(cx+d)^{-1}$ *to the same formulas with new control points* $p'_i = \Phi(p_i)$ *and new weights* $w'_i = \hat{\Phi}(p_i, w_i) = (cp_i + d)w_i$.

8.4.3 $\mathbb{C}$ and $\mathbb{H}$ as Subalgebras of Geometric Algebras

Complex numbers $\mathbb{C}$ are identified with an even subalgebra of $\mathcal{G}_2$:

$$\mathrm{in}_{\mathbb{C}} : \mathbb{C} \to (\mathcal{G}_2)_{\mathrm{even}}, \quad z = x + y\,\mathrm{i} \mapsto x + ye_{12}. \tag{8.23}$$

Multiplying by e_1 from the right, one can get the standard map from $\mathbb{C}$ to $\mathbb{R}^2$:

$$\mathrm{in}_{\mathbb{C}} : \mathbb{C} \to \mathbb{R}^2, \quad z = x + y\,\mathrm{i} \mapsto e_1 \mathrm{in}_{\mathbb{C}}(z) = xe_1 + ye_2.$$

Hence any complex Bézier formula can be mapped to a Clifford–Bézier one

$$e_1 \mathrm{in}_{\mathbb{C}}\big((\textstyle\sum_i p_i w_i B_i)(\sum_i w_i B_i)^{-1}\big) = (\sum_i p'_i w'_i B_i)(\sum_i w'_i B_i)^{-1}, \tag{8.24}$$

where $p'_i = e_1 \mathrm{in}_{\mathbb{C}}(p_i)$, and $w'_i = \mathrm{in}_{\mathbb{C}}(w_i)$.

Quaternions $\mathbb{H}$ are identified with an even subalgebra of $\mathcal{G}_3$:

$$\mathrm{in}_{\mathbb{H}} : \mathbb{H} \to (\mathcal{G}_3)_{\mathrm{even}}, \quad q = r + x\,\mathrm{i} + y\,\mathrm{j} + z\,\mathrm{k} \mapsto r - xe_{23} + ye_{13} - ze_{12}. \tag{8.25}$$

Using duality $X^* = IX$, one can get the standard map from imaginary quaternions Im $\mathbb{H}$ to $\mathbb{R}^3$:

$$x\,\mathrm{i} + y\,\mathrm{j} + z\,\mathrm{k} \mapsto I\ \mathrm{in}_{\mathbb{H}}(q) = xe_1 + ye_2 + ze_3.$$

Hence any Quaternionic-Bézier formula can be mapped to a Clifford–Bézier one

$$I\ \mathrm{in}_{\mathbb{H}}\big((\textstyle\sum_i p_i w_i B_i)(\sum_i w_i B_i)^{-1}\big) = (\sum_i p'_i w'_i B_i)(\sum_i w'_i B_i)^{-1}, \tag{8.26}$$

where $p'_i = I\,\mathrm{in}_{\mathbb{H}}(p_i)$, and $w'_i = \mathrm{in}_{\mathbb{H}}(w_i)$.

Therefore all the results from Sects. 8.2 and 8.3 about QB curves and surfaces are valid for the corresponding CB curves and surfaces in the algebra $\mathcal{G}_3$ generated by the Euclidean space $\mathbb{R}^3$ with signature $(3, 0, 0)$.

There are just a couple differences in the formulas:

- Conjugation $q \mapsto \bar{q}$ in $\mathbb{H}$ should be changed to reversion $x \mapsto \tilde{x}$ in $\mathcal{G}_3$,
- The inversion $q \mapsto -q^{-1}$ in Im $\mathbb{H}$ should be changed to $x \mapsto x^{-1}$ in $\mathbb{R}^3 \subset \mathcal{G}_3$.

8.4.4 Conformal Model of Euclidean Space

We are going to demonstrate how our methods can be applied to representing the usual rational Bézier curves and surfaces in the conformal model.

Consider a pseudo-Euclidean space $\mathbb{R}^5_{4,1}$ and its generated geometric algebra $\mathcal{G}_{4,1}$. The standard basis $\{e_1, \dots, e_5\}$, $e_i \cdot e_i = 1$, $i \neq 5$, $e_5 \cdot e_5 = -1$, will be changed to the following one

$$\{e_0, e_1, e_2, e_3, e_\infty\}, \quad e_0 = (-e_4 + e_5)/2, \quad e_\infty = e_4 + e_5.$$

Define an embedding of Euclidean space $\mathbb{R}^3$ to $\mathbb{R}^5_{4,1}$:

$$\mathrm{conf}(x) = x + \tfrac{1}{2}x^2 e_\infty + e_0 \tag{8.27}$$

to a quadric of null-vectors

$$X \cdot X = 0, \quad X \in \mathbb{R}^5_{4,1}. \tag{8.28}$$

If we expand X in the standard basis $X = \sum_i x_i e_i$ then $X \cdot X = x_1^2 + \cdots + x_4^2 - x_5^2$. Hence the quadric (8.28) defines a 3-sphere S^3 in the affine part $x_5 \neq 0$ of the associated projective space $\mathbb{R}P^4 = P(\mathbb{R}^5_{4,1})$. Actually $\mathrm{conf} : \mathbb{R}^3 \to S^3$ is the inverse of stereographic projection.

Let us apply the machinery we developed. Using the identities $xe_\infty + e_\infty x = 0$ and $e_\infty^2 = 0$, one can modify formula (8.27) as the composition of two fractional-linear functions

$$\mathrm{conf}(x) = x(\tfrac{1}{2}xe_\infty + 1) + e_0 = x(\tfrac{1}{2}e_\infty x + 1)^{-1} + e_0.$$

Hence, the map $\mathrm{conf} : \mathbb{R}^3 \to \mathbb{R}^5_{4,1}$ is the restriction of a Möbius transformation $\Phi_C \in \mathrm{M\ddot{o}b}(\mathbb{R}^5_{4,1})$ given by the matrix:

$$C = \begin{pmatrix} 1 + \frac{1}{2}e_0 e_\infty & e_0 \\ \frac{1}{2}e_\infty & 1 \end{pmatrix} = \begin{pmatrix} 1 & e_0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{1}{2}e_\infty & 1 \end{pmatrix}.$$

Therefore, according to Proposition 8.6 one can 'lift' any CB-curve or surface (including the usual rational Bézier curves and surfaces with real weights) to the conformal model. Indeed new control points P_i and weights W_i are related to the old ones p_i and w_i as follows:

$$P_i = \mathrm{conf}(p_i), \quad W_i = (\tfrac{1}{2}p_i e_\infty + 1)w_i. \tag{8.29}$$

The main advantage of this conformal model (i.e. $\mathbb{R}^3$ embedded into $\mathbb{R}^5_{4,1}$) is in the possibility to represent important geometric objects and transformations in $\mathbb{R}^3$ as formulas in the algebra $\mathcal{G}_{4,1}$ (see e.g. [7,8,16]). For example, the conformal image of a circle $C \subset \mathbb{R}^3$ going through three points p_i, $i = 0, 2, 3$, is the intersection of S^3: $X \cdot X = 0$ with a 2-plane $\Pi \wedge X = 0$, where $\Pi = \mathrm{conf}(p_0) \wedge \mathrm{conf}(p_1) \wedge \mathrm{conf}(p_2)$ is a 3-vector. We treat the 3-vector $\Pi \in \mathcal{G}_3$ as a 2-plane and say that it is *associated* with the circle C.

Let us demonstrate how this technique can help us to find a quadratic cone containing the conformal image $\mathrm{conf}(P(s,t))$ of a bilinear CB-patch (cf. Sect. 8.3.3). Using formulas (8.29) we lift the control points p_i and weights w_i given by (8.15) to the corresponding control points P_i and weights W_i in the conformal model. We also compute tangent vectors $V_{ij} = (P_j - P_i)W_j W_i^{-1}$ (see (8.9)). Then the 3-vectors $\Pi_{01} = P_0 \wedge P_1 \wedge V_{01}$, $\Pi_{23} = P_2 \wedge P_3 \wedge V_{23}$ represent 2-planes associated with the opposite boundary circles C_{01} and C_{23} from the same family (in the notation of Lemma 8.3). The family of 2-planes associated with the paired family of circles can be obtained using the classical Steiner construction by intersecting two pencils of hyperplanes defined by the 2-planes Π_{01} and Π_{23}. We have two obvious corresponding pairs of hyperplanes in these pencils: $\Pi_{01} \wedge V_{02}, \Pi_0 \wedge P_2$ and $\Pi_{23} \wedge P_0, \Pi_2 \wedge V_{20}$. Hence the implicit equation of the quadric cone we are looking for should be the determinant of these four hyperplanes:

$$(\Pi_{01} \wedge V_{02} \wedge X)(\Pi_{23} \wedge V_{20} \wedge X) - (\Pi_{01} \wedge P_2 \wedge X)(\Pi_{23} \wedge P_0 \wedge X) = 0.$$

Of course in order to fix a correct projective correspondence between the pencils one needs three corresponding hyperplanes on these pencils. So in general a certain additional coefficient will be needed in the above equation. In our case the coefficient is 1 by the magic of geometric algebra.

8.4.5 CB-Surfaces in Isotropic Space and PN-Surfaces

In this Section we survey results of [15] on CB-surfaces based on the geometric algebra $\mathcal{G}_{2,0,1}$ generated by an *isotropic space* $\mathbb{R}^3_{2,0,1}$.

The signature $\sigma = (2, 0, 1)$ means that $x \cdot x = x_1^2 + x_2^2$ in coordinates of the standard basis (see (8.19)). Therefore, distances in *isotropic geometry* are measured as Euclidean distances in the projection to the first two coordinates, which is called a *top view*. *Isotropic Möbius* (i-M) transformations are elements of the group $\mathrm{Möb}(\mathbb{R}^3_{(2,0,1)})$ as defined in Sect. 8.4.2. The distinguished vertical direction separates all planes into two classes: vertical (isotropic) and non-vertical planes. Images of these two classes of planes under i-M transformations generate two types of *isotropic spheres* (i-spheres): paraboloids of revolution with a vertical axis (parabolic type) and cylinders with top view circles (cylindrical type). An *isotropic* *circle* (i-circle) is the intersection between an i-sphere of parabolic type and a plane: it is either an ellipse with a circle as top view or a parabola with a vertical axis.

It appears that the theory of QB-surfaces developed in Sects. 8.2 and 8.3 (and initially introduced in [13]) can be successfully developed in the case of CB-surfaces in isotropic space. In general one just needs to add everywhere 'isotropic', e.g. circles and M-transformations should be changed to i-circles and i-M-transformations. The counterpart of a Dupin cyclide is an *isotropic cyclide*: a quartic surface in $\mathbb{R}^3_{2,0,1}$ with a double conic $x_1^2 + x_2^2 = 0$ at infinity or its i-M-transformations.

In [15] bilinear CB-patches in $\mathbb{R}^3_{2,0,1}$ are studied in much detail: their implicitization formula is derived, they are characterized as patches on isotropic cyclides and the uniqueness of patches with three given boundary isotropic circles is proved.

The motivation for these studies is in the following theorem due to Pottmann and Peternell [17, 18]:

Theorem 8.5 *The duality* (8.30) *defines a 1–1 correspondence between non-developable PN-surfaces in the Euclidean space* $\mathbb{R}^3$ *and rational surfaces in the isotropic space* $\mathbb{R}^3_{2,0,1}$.

We recall here PN-surfaces and the construction of duality. *Pythagorean-normal* (PN) surfaces are rational surfaces in the Euclidean space $\mathbb{R}^3$ together with a field of rational unit normals. PN-surfaces are important in geometric modeling applications, since they are rational surfaces with rational offsets. Following [17, 18] (see also survey in [12]), we map oriented planes in $\mathbb{R}^3$ to points of the isotropic space

$$n_1x_1 + n_2x_2 + n_3x_3 + h = 0 \mapsto \frac{1}{n_3 + 1}(n_1, n_2, h) \in \mathbb{R}^3_{2,0,1}. \tag{8.30}$$

Treating a surface in $\mathbb{R}^3$ as the set of its oriented tangent planes, then applying the map (8.30) we get a *dual* surface in $\mathbb{R}^3_{2,0,1}$

We end this exposition with one PN-surface modeling example.

Example 8.2 Consider a corner defined by three orthogonal planes, where three edges are blended using cylinders of radii $r_1 < r_2 < r_3$. The goal is to find a quadrangular PN-patch that will blend smoothly the given three cylinders and the top horizontal plane as shown in Fig. 8.5. The idea is to apply duality (8.30): the cylinders go to i-circular arcs and the top plane goes to a point in the isotropic space. Using the isotropic analog of Lemma 8.3 (see [15, Theorem 1]), one can fill the triangular contour shown in Fig. 8.5 (left) with a bilinear CB-patch and go back using duality. The resulting PN-surface patch can be represented as a Bézier surface of bidegree (3, 4), Fig. 8.5 (right).

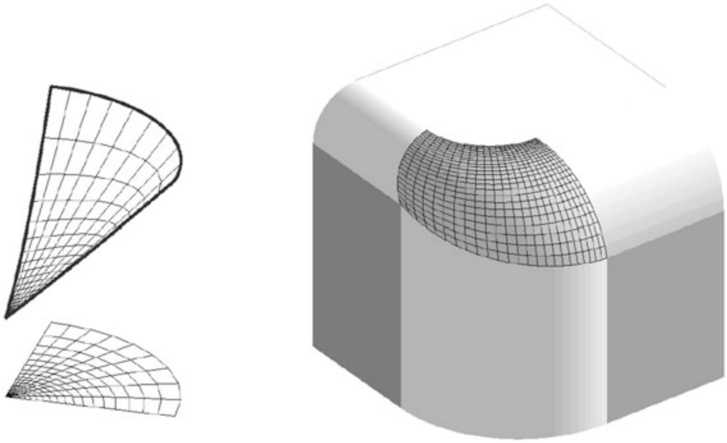


Fig. 8.5 A CB-patch (with a top view) and its dual PN-surface patch

Conclusions

We introduced quaternionic Bézier curves and surfaces in Euclidean space $\mathbb{R}^3$ with two main advantages compared to the customary rational Bézier case: more compact representation (the degree is halved) and Möbius invariance. Disadvantages include absence of affine invariance and complicated conditions on control points and weights that keep quaternionic Bézier curves and surfaces in $\mathbb{R}^3$.

The simplest non-trivial case of bilinear quaternionic Bézier patches was studied:

- The implicitization formula is presented;
- They are characterized as Darboux cyclide patches;
- All such patches on a given Darboux cyclide are classified;
- Principal patches on Dupin cyclides are presented in this form.

We also have shown that complex or quaternionic Bézier formulas can be translated to more general geometric algebra settings. This approach is useful for representing usual rational Bézier curves and surfaces in the conformal model, and for understanding bilinear Clifford–Bézier patches in isotropic space, which have potential applications to rational offset surface modeling.

Acknowledgements The authors would like to thank Helmut Pottmann for pointing out that general bilinear quaternionic patches may represent Darboux cyclides and providing access to the preliminary version of [19]. The majority of the numerical experiments and symbolic computations were made for this paper using the software package CLUCalc/CLUViz described in [16] and the MAPLE package Clifford [1].

References

1. R. Ablamowicz, B. Fauser, A Maple 10 Package for Clifford Algebra Computations, Version 10 (2007). http://math.tntech.edu/rafal/cliff10
2. G. Albrecht, W.L.F. Degen, Construction of Bézier rectangles and triangles on the symmetric Dupin horn cyclide by means of inversion. Comput. Aided Geom. Des. **14**, 349–375 (1997)
3. C. Bisi, G. Gentili, Moebius transformations and the poincare distance in the quaternionic setting. Indiana Univ. Math. J. **58**, 2729–2764 (2010)
4. A. Bobenko, U. Pinkall, Discrete isothermic surfaces. J. Reine Angew. Math. **475**, 187–208 (1996)
5. W. Degen, Cyclides, in *Handbook of Computer Aided Geometric Design* (Elsevier, Amsterdam/Boston, 2002), pp. 575–601
6. R. Dietz, J. Hoschek, B. Juettler, An algebraic approach to curves and surfaces on the sphere and on other quadrics. Comput. Aided Geom. Des. **10**, 211–229 (1993)
7. L. Dorst, D. Fontijne, S. Mann, *Geometric Algebra for Computer Science* (Morgan-Kaufmann, San Francisco, 2007)
8. R. Goldman, A Homogeneous model for three-dimensional computer graphics based on the clifford algebra for $\mathbb{R}^3$, in *Guide to Geometric Algebra in Practice*, ed. by L. Dorst, J. Lasenby (Springer, New York, 2011), pp. 329–352
9. V. Karpavičius, R. Krasauskas, Real-time visualization of Möbius transformations in space using Quaternionic-Bézier approach, in *21-st International Conference on Computer Graphics, Visualization and Computer Vision (WSCG), Communication Papers Proceedings*, Pilsen, 2013, pp. 259–266. http://wscg.zcu.cz/wscg2013/program/short/C17-full.pdf
10. R. Krasauskas, Bézier patches on almost toric surfaces, in *Algebraic Geometry and Geometric Modeling* (Springer, Berlin/New York, 2006), pp. 135–150
11. R. Krasauskas, C. Maeurer, Studying cyclides with laguerre geometry. Comput. Aided Geom. Des. **17**, 101–126 (2000)
12. R. Krasauskas, M. Peternell, Rational offset surfaces and their modeling applications, in *IMA Volume 151: Nonlinear Computational Geometry*, ed. by I.Z. Emiris, F. Sottile, T. Theobald (Springer, New York, 2010), pp. 109–135
13. R. Krasauskas, S. Zube, Bézier-like parametrizations of spheres and cyclides using geometric algebra, in *Proceedings of 9th International Conference on Clifford Algebras and Their Applications in Mathematical Physics*, Weimar, 2011, ed. by K. Guerlebeck
14. R. Krasauskas, S. Zube, Representation of Dupin cyclides using quaternions (in preparation)
15. R. Krasauskas, S. Zube, S. Cacciola, Bilinear Clifford-Bézier patches on isotropic cyclides, in *Mathematical Methods for Curves and Surfaces*. LNCS, vol. 8177, ed. by M. Floater et al. (Springer, Berlin/Heidelberg, 2014), pp. 283–303
16. C. Perwass, *Geometric Algebra with Applications in Engineering*. Series: geometry and computing, vol. 4 (Springer, Berlin/Heidelberg, 2009)
17. M. Peternell, H. Pottmann, A Laguerre geometric approach to rational offsets. Comput. Aided Geom. Des. **15**, 223–249 (1998)
18. H. Pottmann, M. Peternell, Applications of Laguerre geometry in CAGD. Comput. Aided Geom. Des. **15**, 165–186 (1998)
19. H. Pottmann, L. Shi, M. Skopenkov, Darboux cyclides and webs from circles. Comput. Aided Geom. Des. **29**, 77–97 (2012)
20. J. Sanchez-Reyes, Complex rational Bézier curves. Comput. Aided Geom. Des. **26**, 865–876 (2009)
21. S. Zube, A circle representation using complex and quaternion numbers. Lith. J. Math. **46**, 298–310 (2006)
22. S. Zube, Quaternionic rational Bézier curves. Lietuvos matematikos rinkinys **52**, 53–58 (2011)

Part III
Algebraic Geometry for CAD Applications

Chapter 9
Algebraic Spline Geometry: Some Remarks

Ragni Piene

9.1 Algebraic Geometry and Geometric Modeling

Applications of algebraic geometry to Computer Aided Geometric Design often rely on mathematical foundations based on the use of complex numbers and projective geometry. However, all the considered practical questions are presented over the real numbers and in an affine setting. This obvious remark implies that one needs to focus on the study of *real algebraic geometry* and consider curves, surfaces and solids as semi-algebraic sets (i.e., sets defined by means of polynomial equalities and inequalities).

Certain aspects of the foundations of algebraic geometry, with special emphasis on classical projective geometry of curves and surfaces, need to be extended and developed for the real, affine, bounded – and in particular polyhedral – cases, having in mind applications to CAGD. Particular issues that could be considered are singularity theory: existence and description of real surface singularities; the theory of polar and dual varieties in order to relate the description of real polar varieties to Sturm–Habicht methods for determining the topology of real surfaces, and to find efficient ways of computing points on the components of real varieties [5, 6]; the theory of moduli spaces of varieties and of parameterized varieties and the study of the semi-algebraic stratification of these moduli spaces; the design of parametric catalogues of surfaces (parametric or not) that can be used in CAGD.

When modeling curves and surfaces algebraically, using just one interpolating polynomial does not necessarily give a good approximation, due to Runge's phenomenon, even if the polynomial has high degree. It is preferable to use

R. Piene (✉)
CMA/Department of Mathematics, University of Oslo, P. O. Box 1053 Blindern,
NO-0316 Oslo, Norway
e-mail: ragnip@math.uio.no

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_9

piecewise polynomials to approximate larger regions of a CAGD-model. By using more polynomials, one can keep the polynomial degree low and still get better approximations. An *algebraic spline function* is simply a piecewise polynomial function on a polyhedral subdivision of a region in $\mathbb{R}^d$.

By replacing affine, real algebraic varieties in $\mathbb{R}^d$ by simplicial complexes Δ embedded in $\mathbb{R}^d$, and polynomial functions by algebraic spline functions, one gets a theory that we can call *algebraic spline geometry*. The purpose of such a theory is to establish a better basis for the application of multivariate algebraic splines to problems in geometric modeling. To bring real algebraic geometry and spline surface representation in CAGD closer together, we should also develop and understand certain aspects of the "classical" semi-algebraic real geometry so that it can extend to the theory of multivariate algebraic splines.

A simplicial complex in $\mathbb{R}^d$ can be thought of as a union of semi-algebraic sets, each determined by linear equations and linear inequalities, fitting together in a prescribed way. One can generalize this approach by using polynomials of arbitrary degree, as is done in [9], and to a certain extent in [12]. Still another point of view is the following (cf. [11]): replace ordinary algebraic sets in $\mathbb{R}^d$ (defined as zero sets of polynomials) by "piecewise" algebraic sets defined as zero sets of algebraic splines on some given domain in $\mathbb{R}^d$.

9.2 Algebraic Spline Spaces

Consider a (pure) d-dimensional simplicial complex Δ in $\mathbb{R}^d$. We let $C^r(\Delta)$ denote the set of piecewise polynomials (algebraic splines) on Δ of smoothness r. This set is a ring under the usual pointwise addition and multiplication. The (global) polynomial functions $\mathbb{R}[x_1,\dots,x_d]$ are of course r-smooth for any r, and hence can be considered as a subring of every $C^r(\Delta)$. These functions will be called the *trivial splines*.

The subsets $C^r_k(\Delta) \subset C^r(\Delta)$ consisting of splines of degree $\leq k$ are $\mathbb{R}$-vector spaces, called *algebraic spline spaces*, and a major problem is to determine their dimension. In addition, one would like to find explicit bases for these spaces.

In the context of the SAGA project, Mourrain and Villamizar addressed the first problem in the case of $d = 2$ [7] and $d = 3$ [10, Ch. 3]. Using the homological approach introduced by Billera [1] and further developed by Billera–Rose [2, 3], they gave upper and lower bounds for the dimension of the vector spaces $C^r_k(\Delta)$.

Yuzvinsky [13] considered $C^r(\Delta)$ as the global sections of a sheaf $\mathcal{F}^r$ on the poset associated to Δ. He was, like Billera and Rose, interested in the structure of $C^r(\Delta)$ as a module over the ring of trivial splines, for example finding conditions for it to be a free module or a Cohen–Macaulay module, and conditions for the sheaf $\mathcal{F}^r$ to be flasque.

9.3 Generalized Stanley–Reisner Rings

Let $\varDelta$ be a simplicial complex in $\mathbb{R}^d$, with vertices $\{v_1, \dots, v_n\}$. The face ring, or Stanley–Reisner ring of $\varDelta$, is the ring $A_\varDelta := \mathbb{R}[Y_1, \dots, Y_n]/I_\varDelta$, where $I_\varDelta$ is the monomial ideal generated by the products $Y_{i_1} \cdots Y_{i_j}$ such that $\{v_{i_1}, \dots, v_{i_j}\}$ is not a face of $\varDelta$. It is known [4] that if two simplicial complexes have isomorphic Stanley–Reisner rings, then they are themselves isomorphic.

By identifying Y_i with the Courant function $Y_i(v_j) = \delta_{ij}$ extended by linearity, we would have $\sum_i Y_i = 1$, and the ring $A_\varDelta/(\sum Y_i - 1)$ can be viewed as the ring $C^0(\varDelta)$ of (continuous) splines on $\varDelta$. This ring is sometimes referred to as the *affine* Stanley–Reisner ring of $\varDelta$. The points of $\mathrm{Spec}(C^0(\varDelta))$ are the points on $\mathrm{Spec}(A_\varDelta) \subset \mathbb{R}^n$ that lie on the hyperplane $\sum_i Y_i = 1$, and those points that have non-negative coordinates, give a model of $\varDelta$.

As observed by Billera and Rose [3], one can "homogenize" the Stanley–Reisner ring by replacing $\varDelta$ by the $(d+1)$-dimensional simplicial complex $\hat{\varDelta}$ obtained by embedding $\varDelta$ in the hyperplane $x_0 = 1$ in $\mathbb{R}^{d+1}$ and taking its cone with vertex at the origin $\hat{v}_0 = (0, \dots, 0)$. Then

$$C^0(\hat{\varDelta}) = \mathbb{R}[\hat{Y}_0, \hat{Y}_1, \dots, \hat{Y}_n]/(I_{\hat{\varDelta}} + (\sum_{i=0}^{n} \hat{Y}_i - 1)) = A_{\hat{\varDelta}}/(\sum_{i=0}^{n} \hat{Y}_i - 1) \cong A_\varDelta,$$

where the variables $\hat{Y}_i$ represent the Courant functions of $\hat{\varDelta}$ (see [8]). Since $I_\varDelta$ is a homogeneous ideal, this ring is graded, and Proj $C^0(\hat{\varDelta})$ also "describes" $\varDelta$.

For $r \geq 0$, consider the subalgebras $C^r(\varDelta) \subseteq C^0(\varDelta) = A_\varDelta/(\sum Y_i - 1)$ and $C^r(\hat{\varDelta}) \subseteq C^0(\hat{\varDelta}) = A_\varDelta$ consisting of splines of smoothness r. We call these rings *generalized Stanley–Reisner rings*. As observed by Billera [1], the space of splines on $\varDelta$ of smoothness r and degree $\leq k$ correspond to the space of splines on $\hat{\varDelta}$ of smoothness r and degree exactly k, in particular the dimensions of these two vector spaces are equal.

Here we concentrate on the "affine" generalized Stanley–Reisner rings $C^r(\varDelta)$. In the following, we compute these rings and explain their geometric realization $\mathrm{Spec}(C^r(\varDelta))$ only in two particularly simple cases. Nevertheless, we think that even these two examples support our conjecture, namely that the part of $\mathrm{Spec}(C^r(\varDelta))$ with non-negative coordinates (with respect to a certain embedding into an affine space) models a twisted version of the simplicial complex $\varDelta$.

Example 1: $d = 1$. Let $\varDelta$ be a one-dimensional simplicial complex with three vertices $v_1, v_2, v_3 \in \mathbb{R}$, and assume $v_1 < v_2 < v_3$ (Fig. 9.1).

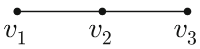


Fig. 9.1 The one-dimensional simplicial complex of Example 1

Since only $\{V_1, V_3\}$ is not a face, we have

$$C^0(\Delta) = A_\Delta/(\sum_{i=1}^{3} Y_i - 1) = \mathbb{R}[Y_1, Y_2, Y_3]/(Y_1Y_3, Y_1 + Y_2 + Y_3 - 1).$$

Then $\mathrm{Spec}(C^0(\Delta)) \subset \mathbb{R}^3$ is the union of the lines $Y_1 = Y_2 + Y_3 - 1 = 0$ and $Y_3 = Y_1 + Y_2 - 1 = 0$. The segments of these two lines contained in the positive octant mimic the two 1-faces of Δ, and they intersect transversally. Set

$$H := v_1Y_1 + v_2Y_2 + v_3Y_3.$$

Then $H(x) = x$ for any point $x \in |\Delta|$. So $\mathbb{R}[H]$ is the subring of trivial splines, and $\mathbb{R}[H] \subseteq C^r(\Delta)$, for any $r \geq 0$.

By performing the (affine) computations similar to what is done in [8], one sees that H, Y_1^{r+1}, and Y_3^{r+1} generate the subring $C^r(\Delta)$ of $C^0(\Delta)$. Note that, since

$$Y_1H = (v_1 - v_2)Y_1^2 + v_2Y_1 \text{ and } Y_3H = (v_3 - v_2)Y_3^2 + v_2Y_3$$

in $C^0(\Delta)$, all powers of Y_1 and Y_3 higher than $r + 1$ are in the subring generated by (the images of) H, Y_1^{r+1}, and Y_3^{r+1}. Now consider the homomorphism

$$\mathbb{R}[x, y, z] \to \mathbb{R}[Y_1, Y_2, Y_3]/(Y_1Y_3, Y_1 + Y_2 + Y_3 - 1)$$

that sends x to $H - v_2$, y to $(v_1 - v_2)^{r+1}Y_1^{r+1}$, and z to $(v_3 - v_2)^{r+1}Y_3^{r+1}$. The image of this map is $C^r(\Delta)$ and the kernel is the ideal $(yz, y + z - x^{r+1})$. Hence

$$C^r(\Delta) \cong \mathbb{R}[x, y, z]/(yz, y + z - x^{r+1}),$$

and $\mathrm{Spec}(C^r(\Delta))$ is the union of the plane curves $y = z - x^{r+1} = 0$ and $z = y - x^{r+1} = 0$. For $r \geq 1$, these two curves both have the x-axis as tangent at their point of intersection (the origin), and the tangent intersects each curve with multiplicity $r + 1$ (Fig. 9.2).

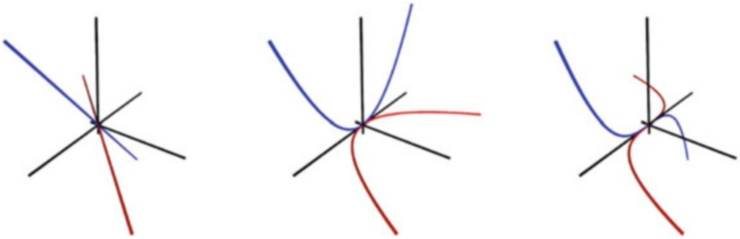


Fig. 9.2 $\mathrm{Spec}(C^r(\Delta))$ for $r = 0, 1, 2$

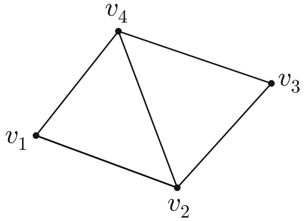


Fig. 9.3 The two-dimensional simplicial complex of Example 2

Example 2: $d = 2$. Consider a two-dimensional simplicial complex Δ, with four vertices $v_1, \dots, v_4 \in \mathbb{R}^2$, no three on a line, and $\{v_1, v_3\}$ as the only non-face. Write $v_i = (v_{i1}, v_{i2})$ (Fig. 9.3).

Then

$$C^0(\Delta) = \mathbb{R}[Y_1, Y_2, Y_3, Y_4]/(Y_1Y_3, Y_1 + Y_2 + Y_3 + Y_4 - 1),$$

where the variables Y_i represent the Courant functions. The trivial splines in this case are

$$H_j := v_{1j}Y_1 + v_{2j}Y_2 + v_{3j}Y_3 + v_{4j}Y_4, \text{ for } j = 1, 2.$$

As in the previous example, we observe that Y_1^{r+1} and Y_3^{r+1} are in $C^r(\Delta)$. Moreover, we deduce the following linear relation (assuming for simplicity that none of the denominators are zero):

$$\frac{H_1 - v_{21}}{v_{41} - v_{21}} - \frac{H_2 - v_{22}}{v_{42} - v_{22}} = \Big(\frac{v_{11} - v_{21}}{v_{41} - v_{21}} - \frac{v_{12} - v_{22}}{v_{42} - v_{22}}\Big)Y_1 + \Big(\frac{v_{31} - v_{21}}{v_{41} - v_{21}} - \frac{v_{32} - v_{22}}{v_{42} - v_{22}}\Big)Y_3.$$

Multiplying this relation by powers of Y_1 and of Y_3, we see that the subring of $C^r(\Delta)$ generated by the images of $H_1, H_2, Y_1^{r+1}, Y_3^{r+1}$ contain all powers of Y_1 and Y_3 higher than $r+1$. Hence we get that $C^r(\Delta)$ is the image of the homomorphism

$$\mathbb{R}[x_1, x_2, y, z] \to \mathbb{R}[Y_1, Y_2, Y_3, Y_4]/(Y_1Y_3, Y_1 + Y_2 + Y_3 + Y_4 - 1),$$

sending x_1 to $\frac{H_1 - v_{21}}{v_{41} - v_{21}}$, x_2 to $\frac{H_2 - v_{22}}{v_{42} - v_{22}}$, y to $\big(\frac{v_{11} - v_{21}}{v_{41} - v_{21}} - \frac{v_{12} - v_{22}}{v_{42} - v_{22}}\big)^{r+1} Y_1^{r+1}$, and z to $\big(\frac{v_{31} - v_{21}}{v_{41} - v_{21}} - \frac{v_{32} - v_{22}}{v_{42} - v_{22}}\big)^{r+1} Y_3^{r+1}$. The kernel of this homomorphism is generated by yz and $y + z - (x_1 - x_2)^{r+1}$, so that

$$C^r(\Delta) \cong \mathbb{R}[x_1, x_2, y, z]/(yz, y + z - (x_1 - x_2)^{r+1}).$$

Hence $\operatorname{Spec}(C^r(\Delta))$ is the union of the surfaces $y = z - (x_1 - x_2)^{r+1} = 0$ and $z = y - (x_1 - x_2)^{r+1} = 0$. The intersection of these surfaces is the line $y = z = x_1 - x_2 = 0$. The plane $y = z = 0$ is the tangent plane to both surfaces at all points of their line of intersection. The intersection of this tangent plane and each surface is the line, with multiplicity $r+1$.

If we instead consider the "homogeneous" Stanley–Reisner ring $C^0(\hat{\Delta}) \cong A_\Delta$, we set $H_0 := Y_1 + Y_2 + Y_3 + Y_4$ and define the homomorphism

$$\mathbb{R}[x_0, x_1, x_2, y, z] \to \mathbb{R}[Y_1, Y_2, Y_3, Y_4]/(Y_1 Y_3),$$

by sending x_0 to H_0, X_1 to $\frac{H_1 - v_{21} H_0}{v_{41} - v_{21}}$, x_2 to $\frac{H_2 - v_{22} H_0}{v_{42} - v_{22}}$, and y and z as before. We then get

$$C^r(\hat{\Delta}) \cong \mathbb{R}[x_0, x_1, x_2, y, z]/(yz, y + z - (x_1 - x_2)^{r+1}).$$

This is a graded ring, where the x_i have weight 1 and y and z have weight $r+1$. Hence Conjecture 3.1 of [10, Ch. 3], stated only for $r = 1$ and with a different choice of variables, holds for all r. Note that $\mathrm{Spec}(C^r(\hat{\Delta}))$ is an affine cone, and that we can view $\mathrm{Spec}(C^r(\Delta))$ as the intersection of this cone with the hyperplane $x_0 = 1$.

We see that, as a module over the ring of trivial splines $\mathbb{R}[x_1, x_2]$, $C^r(\Delta)$ is isomorphic to $\mathbb{R}[x_1, x_2] \oplus y\mathbb{R}[x_1, x_2]$ (note that the isomorphism depends on r). Moreover, the r-splines of degree $\leq k$ are of the form $f(x_1, x_2) + yg(x_1, x_2)$, where f is a polynomial of degree $\leq k$ and g is of degree $\leq k - (r+1)$. Hence

$$\dim C^r_k(\Delta) = \binom{k+2}{2} + \binom{k+2-(r+1)}{2},$$

in accordance with [7, Thm. 13, p. 572], since there are no interior vertices in Δ.

It is natural to conjecture that the situation of these examples is the general one, namely that $\mathrm{Spec}(C^r(\Delta))$ represents a twisted version of Δ, in which the varieties corresponding to the d-faces are curved and intersect along varieties corresponding to the $(d-1)$-faces "with multiplicity $r+1$" as described above. More precisely, based on the above two examples, we conjecture the following "local" description of $\mathrm{Spec}(C^r(\Delta))$.

Local Spline Ring Conjecture *Let Δ be a d-dimensional simplicial complex consisting of two d-simplices intersecting in a $(d-1)$-simplex. Then we can realize* $\mathrm{Spec}(C^r(\Delta)) \subset \mathbb{R}^{d+2}$ *as the union of two smooth d-dimensional varieties V_1 and V_2 intersecting along a linear $(d-1)$-dimensional space L, such that V_1 and V_2 have the same d-dimensional linear space T as tangent space at each point of L and such that V_i and T have order of contact* $r+1$ *at each point of L.*

In [10, Ch. 4], Villamizar considered several examples of simplicial complexes Δ in the special case $d = 2$, $r = 1$, and gave conjectural generators for $C^1(\hat{\Delta})$ as a $\mathbb{R}$-subalgebra of $C^0(\hat{\Delta})$. The computations performed there for splines of low degree, using again the method of [8], support the above conjecture.

Acknowledgements I would like to thank Nelly Villamizar and Alicia Dickenstein for helpful discussions. I am very grateful to Georg Muntingh for making the figures. Finally I thank the referees for their useful comments.

References

1. L.J. Billera, Homology of smooth splines: generic triangulations and a conjecture of Strang. Trans. AMS **310**, 325–340 (1988)
2. L.J. Billera, L.L. Rose, Gröbner basis methods for multivariate splines, in *Mathematical Methods in Computer Aided Geometric Design* (Academic, New York, 1989), pp. 93–104
3. L.J. Billera, L.L. Rose, A dimension series for multivariate splines. Discret. Comput. Geom. **6**, 107–128 (1991)
4. W. Bruns, J. Gubeladze, Combinatorial invariance of Stanley–Reisner rings. Georgian Math. J. **3**(4), 315–318 (1996)
5. H. Mork, Real algebraic curves and surfaces, Ph.D. thesis, University of Oslo, 2011
6. H. Mork, R. Piene, Polars of real singular plane curves, in *Algorithms in Algebraic Geometry*, ed. by A. Dickenstein, F.-O. Schreyer, A.J. Sommese. IMA Volumes, vol. 146 (Springer, New York, 2008), pp. 99–115
7. B. Mourrain, N. Villamizar, Homological techniques for the analysis of the dimension of triangular spline spaces. J. Symb. Comput. **50**, 564–577 (2013)
8. H. Schenck, Subalgebras of the Stanley–Reisner ring. Discret. Comput. Geom. **21**, 551–556 (1999)
9. P. Stiller, Certain reflexive sheaves on $\mathbb{P}^n_C$ and a problem in approximation theory. Trans. AMS. **279**, 125–142 (1983)
10. N. Villamizar, Algebraic geometry for splines, Ph.D. thesis, University of Oslo, 2012
11. W.-J. Wang, The irreducibility and isomorphism of piecewise algebraic sets. J. Math. Res. Expo. **24**(2), 225–230 (2004)
12. S. Yuzvinsky, Flasque sheaves on posets and Cohen–Macaulay unions of regular varieties. Adv. Math. **73**, 24–42 (1989)
13. S. Yuzvinsky, Modules of splines on polyhedral complexes Math. Z. **210**, 245–254 (1992)

Chapter 10
On the Dimension of Spline Spaces on Triangulations

Nelly Villamizar and Bernard Mourrain

10.1 Introduction

For a polyhedral partition Δ embedded in $\mathbb{R}^d$, a polynomial *spline* is a piecewise polynomial function defined on Δ with a specified order of global smoothness.

It is commonly accepted that the first mathematical reference to splines is in the article from 1946 by Schoenberg [26], which is probably the first place where the word “spline" is used in connection with smooth, piecewise polynomial approximation. However, the idea that polynomials were the most convenient functions for approximation and interpolation has its roots in the aircraft and shipbuilding industries. On an irregular domain, such as an airplane or a human skull, the approximation functions are needed to be defined and adaptable to satisfy boundary conditions on domains of any reasonable shape. This idea developed into what is now one of the most powerful tools to solve partial differential equations, the finite element method [11]. Splines are nowadays important not only in numerical analysis and approximation theory, they are very useful for modeling surfaces of arbitrary topology and are a widely recognized tool in isogeometric analysis [9], image analysis and free-form representation in Computer Aided Design (CAD) and Computer Aided Geometric Design (CAGD) [10].

N. Villamizar (☒)
Centre of Mathematics for Applications (CMA), University of Oslo, P.O. Box 1053 Blindern
0316 Oslo, Norway

RICAM, Austrian Academy of Sciences, Altenbergerstraße 69, 4040 Linz, Austria
e-mail: nelly.villamizar@oeaw.ac.at

B. Mourrain
INRIA Sophia Antipolis, 2004 route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex,
France
e-mail: Bernard.Mourrain@inria.fr

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_10

To be useful in computations, the space of spline functions must have a basis, and it in turn makes essential to study the dimension of these spaces. For the space of piecewise polynomials on a given triangulation, or on a simplicial partition in $\mathbb{R}^d$, the problem of finding the dimension and a suitable basis for it was first formally formulated by Strang [29, 30]. He conjectured a formula for the dimension of the spline space on a general triangulation [29]. However, serious difficulties already begin to arise in the planar case, and the actual lower bound on the dimension of the space is usually larger than the formula in the conjecture depending on the embedding of the triangulation in $\mathbb{R}^2$ (Morgan and Scott, The dimension of the space of C^1 piecewise polynomials. Unpublished manuscript. http://citeseerx.ist.psu.edu/viewdoc/summary? doi:10.1.1.42.4635, 1975; [19, 27]). The methods to compute the dimension include the construction of nodal bases and the Bernstein–Bézier approach, see [16] and the references therein.

In 1988, Billera introduced the use of homological algebra and some algebraic machinery to study the spaces of splines on triangulations in any dimension [4]. By means of this approach, complicated linear algebra can be presented in a more organized way, and he was able to find the dimension for the space of C^1 bivariate splines for triangulations whose edges are in sufficiently general position, for any fixed polynomial degree. See also [5, 7] for more results concerning the algebraic structure of the spline space.

The homological construction was continued by Schenck and Stillman in [25], and studied in [13, 23–25]. We start the next section by recalling this construction, which we latter use to prove a formula for an upper bound on the dimension of bivariate spline spaces, and new lower and upper bounds for trivariate spline spaces.

These bounds improve previous results [16, 27] and the approach leads to connections of the dimension problem on spline spaces and classical problems in algebraic geometry.

The formula for the upper bound in the bivariate case applies to any ordering established on the interior vertices of the partition. Having no restriction on the ordering makes it possible to obtain accurate approximations to the dimension and even exact values in many cases, for instance it leads to a simple proof for a dimension formula of the C^r spline space when the degree of the polynomials is at least $4r + 1$ [20].

The results for the trivariate case in the literature do not take into account the exact geometry of faces in the partition [1, 2, 17]. The bounds we prove include terms that take into account the geometry of the faces surrounding the interior edges and vertices of the partition. For our results we explore connections between spline functions and ideals generated by powers of linear forms, ideals of fat points, the Fröberg's conjecture and the weak Lefschetz property, giving so an insight into ways of improving these bounds by using results from algebraic geometry [21].

The structure of this chapter is as follows. In Sect. 10.2 we present in detail the construction of the chain complex introduced by Schenck and Stillman [25] in general settings i.e., for any finite d–dimensional simplicial complex. We recall some properties of the homology modules, which leads to a formula for the dimension with terms corresponding to the dimension of low homology modules.

By bounding these terms we get lower and upper bounds for two dimensional simplicial complexes in Sect. 10.3, and the three dimensional case is considered in Sect. 10.4. At the end of each section we present some examples and some final remarks.

10.2 Construction of the Chain Complex

We introduce the notation and some definitions from [23] that will be used throughout this chapter.

We denote by Δ a connected, finite d-dimensional simplicial complex, supported on $|\Delta| \subset \mathbb{R}^d$, such that Δ and all its links are pseudomanifolds [4]. We could think of Δ as the triangulation of a (topological) d-ball.

For integers r and k, with $r \geq 0$, $k \geq r$, denote by $C_k^r(\Delta)$ the space of polynomial splines defined on Δ, of degree at most k, and continuously differentiable of order r.

The problem of finding the dimension of $C_k^r(\Delta)$ can be reduced to the case in which each maximal face of Δ contains the origin in $\mathbb{R}^d$, in the following way [6]. We embed Δ in the hyperplane $\{x_{d+1} = 1\} \subset \mathbb{R}^{d+1}$ and form the cone $\hat{\Delta}$ with vertex at the origin. Let $C^r(\hat{\Delta})$ be the set of splines defined on $\hat{\Delta}$ and continuously differentiable of order r. For a fixed k, let $C^r(\hat{\Delta})_k$ denote the subset of splines on $\hat{\Delta}$ of degree exactly k. The elements of $C^r(\hat{\Delta})_k$ are precisely the homogenization of the elements of $C_k^r(\Delta)$, and

$$C^r(\hat{\Delta})_k \cong C_k^r(\Delta)$$

as $\mathbb{R}$-vector spaces [6]; in particular

$$\dim C_k^r(\Delta) = \dim C^r(\hat{\Delta})_k. \tag{10.1}$$

Thus, we turn the problem of finding $\dim C_k^r(\Delta)$ (for different values of k) into the problem of finding the Hilbert function of a graded algebra, namely $C^r(\hat{\Delta})$, and we may apply the tools of commutative and homological algebra to solve the problem.

Let us denote by Δ^0 the set of interior faces of Δ. For $i = 0, \ldots, d-1$ let Δ_i^0 be the set of i-dimensional interior faces of Δ whose support is not contained in the boundary $\partial\Delta$ of $|\Delta|$, and let Δ_d^0 be the set of all maximal d-faces of Δ. We denote by f_i^0 the cardinality of these sets, for $i = 0, \ldots, d$.

Let us denote by $R := \mathbb{R}[x_1, \ldots, x_{d+1}]$ the polynomial ring in $d+1$ variables. Let $\mathcal{R}$ be the constant (chain) complex on Δ i.e., $\mathcal{R}(\beta) = R$ for every $\beta \in \Delta^0$. For $i = 0, \ldots, d$ we have $\mathcal{R}_i = R^{f_i^0}$. The maps ∂_i in the complex $\mathcal{R}$ are induced by the usual simplicial boundary maps $\bar{\partial}_i$ used to compute the relative homology of $(\Delta, \partial\Delta)$ with coefficients in R (see [4] and [22] for more general details about relative homology).

For $\tau \in \Delta^0_{d-1}$, let ℓ_τ be the linear form that vanishes on $\hat{\tau}$ (it is just the homogenization of the linear polynomial vanishing on τ). For every interior face $\beta \in \Delta^0$ define

$$\mathcal{J}(\beta) := \langle \ell_\tau^{r+1} \rangle_{\tau \ni \beta}.$$

Let us denote by $\mathcal{J}$ the complex of ideals defined in this way, with the restriction of the maps ∂_i defined on $\mathcal{R}_i$ to $\mathcal{J}_i := \bigoplus_{\beta \in \Delta^0_i} \mathcal{J}(\beta)$, for $i = 0, \ldots, d$.

We consider the chain complex $\mathcal{R}/\mathcal{J}$ defined as the quotient of $\mathcal{R}$ by $\mathcal{J}$:

$$0 \xrightarrow{\partial_{d+1}} \bigoplus_{\sigma \in \Delta^0_d} \mathcal{R} \xrightarrow{\partial_d} \bigoplus_{\tau \in \Delta^0_{d-1}} \mathcal{R}/\mathcal{J}(\tau) \xrightarrow{\partial_{d-1}} \cdots \xrightarrow{\partial_1} \bigoplus_{\beta \in \Delta^0_0} \mathcal{R}/\mathcal{J}(\gamma) \xrightarrow{\partial_0} 0$$

where ∂_i are the induced relative (module $\partial\Delta$) simplicial boundary maps.

In [4] it was proved that $C^r(\hat{\Delta})$ is isomorphic to the top homology module of $\mathcal{R}/\mathcal{J}$, i.e.,

$$H_d(\mathcal{R}/\mathcal{J}) := \ker(\partial_d) \cong C^r(\hat{\Delta}).$$

This together with (10.1) and the Euler characteristic equation [28, p. 172]

$$\chi(H(\mathcal{R}/\mathcal{J})) = \chi(\mathcal{R}/\mathcal{J}),$$

implies that, for any fixed $k \geq 1$

$$\dim C^r_k(\Delta) = \sum_{i=0}^{d} (-1)^i \sum_{\beta \in \Delta^0_{d-i}} \dim \mathcal{R}/\mathcal{J}(\beta)_k - \sum_{i=1}^{d} (-1)^i \dim H_{d-i}(\mathcal{R}/\mathcal{J})_k. \tag{10.2}$$

The subindex k indicates that we are considering the k-the part of the graded module.

The aim is to determine a formula for all the modules in the latter equation in terms of known information about Δ.

One first thing that we can use is the short exact sequence

$$0 \longrightarrow \mathcal{J} \longrightarrow \mathcal{R} \longrightarrow \mathcal{R}/\mathcal{J} \longrightarrow 0.$$

It gives rise to the long exact sequence of homology modules

$$\cdots \to H_{i+1}(\mathcal{R}/\mathcal{J}) \to H_i(\mathcal{J}) \to H_i(\mathcal{R}) \to H_i(\mathcal{R}/\mathcal{J}) \to H_{i-1}(\mathcal{J}) \to \cdots \tag{10.3}$$

When Δ is supported on a topological d-ball (as is the case for our simplicial complexes), then $H_i(\mathcal{R}) = 0$ for every $i \neq d$ and $H_d(\mathcal{R}) = R$ [22, p. 181]. Then from the long exact sequence (10.3) it follows that $H_0(\mathcal{R}/\mathcal{J}) = 0$, and for all $i \leq d-1$

$$H_i(\mathcal{R}/\mathcal{J}) \cong H_{i-1}(\mathcal{J}). \tag{10.4}$$

In particular, we get the short sequence

$$0 \longrightarrow H_d(\mathcal{R}) \longrightarrow H_d(\mathcal{R}/\mathcal{J}) \longrightarrow H_{d-1}(\mathcal{J}) \longrightarrow 0,$$

and it follows

$$C^r(\hat{\Delta}) \cong R \oplus H_{d-1}(\mathcal{J}), \tag{10.5}$$

therefore, the study of the spline space reduces to the study of the homology module $H_{d-1}(\mathcal{J})$.

Let us recall that by definition, for $i = 0, \ldots, d$

$$H_i(\mathcal{J}) := \ker(\partial_i)/\operatorname{Im}\partial_{i+1} \tag{10.6}$$

and we have

$$\bigoplus_{\beta \in \Delta_i^0} \mathcal{J}(\beta) = \ker \partial_i \oplus \operatorname{Im} \partial_i, \tag{10.7}$$

where ∂_i are the maps in the chain complex $\mathcal{J}$.

The previous construction is valid in any dimension d, in particular for $d = 2$ and $d = 3$ that are the cases we want to explore here. The importance of the study of ideals of powers of linear forms is easily detectable.

Since $\mathcal{J}(\beta) = 0$ for all maximal faces β of Δ, then

$$\bigoplus_{\beta \in \Delta_d^0} \mathcal{R}/\mathcal{J}(\beta)_k = \bigoplus_{\beta \in \Delta_d^0} \mathcal{R}_k \quad \text{and hence} \quad \dim \bigoplus_{\beta \in \Delta_d^0} \mathcal{R}_k = f_d^0 \cdot \binom{k+d}{d}. \tag{10.8}$$

Also by definition $\mathcal{J}(\beta) = \langle \ell_\beta^{r+1} \rangle$, the ideal generated by the power $r+1$ of the linear form that vanishes on $\hat{\beta}$, for all $\beta \in \Delta_{d-1}^0$, thus

$$\dim \bigoplus_{\beta \in \Delta_{d-1}^0} \mathcal{R}/\mathcal{J}(\beta)_k = f_{d-1}^0 \cdot \left[\binom{k+d}{d} - \binom{k+d-(r+1)}{d} \right]. \tag{10.9}$$

Here and throughout the chapter, we adopt the convention that for $m, u \in \mathbb{Z}$ the binomial coefficient $\binom{m}{u} = 0$ if $m < u$.

Let us consider $\beta \in \Delta_i^0$ for some $0 \leq i < d-1$. Observe that for a specific face β, we may make an affine change of coordinates and assume that the linear forms in $\mathcal{J}(\beta)$ involve only the variables $x_1, \ldots, x_{d-i}$. Hence

$$\mathcal{R}/\mathcal{J}(\beta) \cong \mathbb{R}[x_{d+1-i}, \ldots, x_{d+1}] \otimes_{\mathbb{R}} \mathbb{R}[x_1, \ldots, x_{d-i}]/\mathcal{J}(\beta).$$

Then in order to obtain the dimension of a spline space we need to analyze ideals generated by powers of linear forms in two, three,… and d-variables.

For instance, for a triangulation of a region in the plane, the ideals associated to the vertices ($\beta \in \Delta_0^0$, $d = 2$) are generated by linear forms in two variables; similarly the ideals corresponding to edges in a 3-dimensional partition ($\beta \in \Delta_1^0$, $d = 3$).

Schenck and Stillman [25] proved the following free resolution of ideals in two variables generated by powers of homogeneous linear forms (in [13] Geramita and Schenck extended this result by using *inverse systems of fat points*, they gave a completely characterization of the possible free resolutions for these kind of ideals allowing mixed powers).

Let $\mathcal{J}(\beta)$ be an ideal generated by $\ell_1^{r+1}, \ldots, \ell_t^{r+1}$ where ℓ_j for $j = 1, \ldots, t$ are linearly independent homogeneous linear forms in $\mathbb{R}[x_1, x_2]$. A free resolution of $\mathcal{R}/\mathcal{J}(\beta)$ is given by

$$0 \to \mathcal{R}(-\Omega-1)^a \oplus \mathcal{R}(-\Omega)^b \to \oplus_{j=1}^t \mathcal{R}(-r-1) \to \mathcal{R} \to \mathcal{R}/\mathcal{J}(\beta) \to 0 \tag{10.10}$$

where $\Omega - 1$ is the socle degree of $\mathcal{R}/\mathcal{J}(\beta)$; Ω and the multiplicities a and b are given by

$$\Omega = \left\lfloor \frac{t\,r}{t-1} \right\rfloor + 1, \quad a = t\,(r+1) + (1-t)\,\Omega, \quad b = t - 1 - a. \tag{10.11}$$

Thus, for a fixed polynomial degree k:

$$\dim \mathcal{R}/\mathcal{J}(\beta)_k = \binom{k+d}{d} - t\binom{k+d-(r+1)}{d} + b\binom{k+d-\Omega}{d} + a\binom{k+d-(\Omega+1)}{d}. \tag{10.12}$$

Considering each face β at a the time, the dimension of $\mathcal{R}/\mathcal{J}(\beta)_k$ is given by the previous formula. Summing them up we get the a dimension formula for the corresponding module in (10.2).

For ideals generated by powers of linear forms in three or more variables there is no resolution known. There is a formula conjectured by Fröberg [12] on the expected Hilbert series of an ideal generated by a generic set of forms (not necessarily powers of linear forms) in a polynomials ring in n-variables. Fröberg's conjecture has been proved true in some cases, for instance for $n = 2$, that is the generic situation of the module in (10.12), for $n = 3$ [3]. For other values of n, it has been proved for some particular cases that depend on the number of generators, see [15] for a detailed list. By using Fröberg's conjecture and its equivalent form known as the *maximal rank property*, one might find approximations for the dimension of the spline space, see the discussion in [21] where the case of tetrahedral partitions in $\mathbb{R}^3$ is studied. In Sect. 10.4 below, with $n = 3$, we use a special case for which Fröberg's sequence gives the exact dimension of the ideal and prove a lower bound on the dimension of the spline space $C_k^r(\Delta)$.

As the dimension of the simplicial complexes increases, the homology modules become quite complicated. In Sect. 10.3 we consider splines spaces defined on triangulations embedded in $\mathbb{R}^2$ ($d = 2$), and in Sect. 10.4 spline spaces on simplicial complexes embedded in $\mathbb{R}^3$ ($d = 3$). We find lower and upper bounds for the homology modules and manipulating the formula (10.2) we deduce lower and upper bounds for the dimension of the spline space $C_k^r(\Delta)$.

10.3 Dimension of Bivariate Triangular Spline Spaces

Let Δ be a connected, finite two-dimensional simplicial complex supported on $|\Delta| \subset \mathbb{R}^2$ which is homotopy equivalent to a disk. Applying (10.4), the fomula in (10.2) reduces to

$$\dim C_k^r(\Delta) = \sum_{i=0}^{2}(-1)^i \sum_{\beta\in\Delta_{2-i}^0} \dim \mathcal{R}/\mathcal{J}(\beta)_k + \dim H_0(\mathcal{J})_k. \tag{10.13}$$

Let us recall that in this setting the ring R is the polynomial ring in three variables $R = \mathbb{R}[x_1, x_2, x_3]$, corresponding to the homogenization $\hat{\Delta}$ of Δ.

Theorem 10.1 *The dimension of $C_k^r(\Delta)$ is bounded below by*

$$\dim C_k^r(\Delta) \geq \binom{k+2}{2} + f_1^0\binom{k+1-r}{2}$$
$$-\sum_{i=1}^{f_0^0}\left[t_i\binom{k+1-r}{2} - b_i\binom{k+2-\Omega_i}{2} - a_i\binom{k+1-\Omega_i}{2}\right],$$

where t_i is the number of different slopes of the edges containing the vertex γ_i, and

$$\Omega_i = \left\lfloor \frac{t_i\, r}{t_i - 1} \right\rfloor + 1, \quad a_i = t_i\,(r+1) + (1-t_i)\,\Omega_i, \quad b_i = t_i - 1 - a_i .$$

Proof Since $\dim H_0(\mathcal{J})_k \geq 0$, and (10.8), (10.9) and (10.12) give us formulas for the remaining terms of (10.13), we get the lower bound given in the theorem.

In contrast with the formula of the lower bound, the theorem below provides an upper bound on $\dim C^r_k(\Delta)$. The ordering on the vertices plays an important role in the formula. Different orderings can give different upper bounds, and since the theorem can be applied to any ordering on the vertices in Δ^0_0, it leads to find the exact dimension in many cases [20], see Example 10.1 below.

Let $\gamma_1, \ldots, \gamma_{f^0_0}$ be an ordering on Δ^0_0. For each vertex γ_i, denote by $N(\gamma_i)$ the set of edges that contain γ_i, and define $\tilde{t}_i$ as the number of different slopes of the edges connecting γ_i to a vertex on the boundary or to one of the first $i-1$ vertices in the list.

Theorem 10.2 *The dimension of $C^r_k(\Delta)$ is bounded above by*

$$\dim C^r_k(\Delta) \leq \binom{k+2}{2} + f^0_1 \binom{k+1-r}{2}$$
$$- \sum_{i=1}^{f^0_0} \left[\tilde{t}_i \binom{k+1-r}{2} - \tilde{b}_i \binom{k+2-\tilde{\Omega}_i}{2} - \tilde{a}_i \binom{k+1-\tilde{\Omega}_i}{2} \right]$$

with $\tilde{t}_i$ as we have defined above and

$$\tilde{\Omega}_i = \left\lfloor \frac{\tilde{t}_i\, r}{\tilde{t}_i - 1} \right\rfloor + 1, \quad \tilde{a}_i = \tilde{t}_i\,(r+1) + (1-\tilde{t}_i)\,\tilde{\Omega}_i, \quad \tilde{b}_i = \tilde{t}_i - 1 - \tilde{a}_i .$$

If $\tilde{t}_i = 1$ or 0, then $\tilde{a}_i = \tilde{b}_i = \tilde{\Omega}_i = 0$.

Proof In the case $d = 2$, the isomorphism in (10.5) gives

$$\dim C^r_k(\Delta) = \dim R_k + \dim H_1(\mathcal{J})_k .$$

We know that $H_1(\mathcal{J}) = \ker \partial_1$ by (10.6), and $\bigoplus_{\tau \in \Delta^0_1} \mathcal{J}(\tau) = \ker(\partial_1) \oplus \mathrm{Im}\,(\partial_1)$ by (10.7). Hence we can write

$$\dim C^r_k(\Delta) = \dim R_k + \sum_{\tau \in \Delta^0_1} \dim \mathcal{J}(\tau)_k - \dim(\mathrm{Im}\,\partial_1)_k$$

where ∂_1 is the map in the chain complex $\mathcal{J}$. Therefore, to find an upper bound on $\dim C_k^r(\Delta)$ it is enough to find a lower bound on the dimension of $\operatorname{Im}\partial_1$ in degree k. We define the maps δ, φ and π by R-linear extensions as follows.

Let us consider

$$\delta : \bigoplus_{\tau=(\gamma,\gamma')\in\Delta_1^0} \mathcal{J}(\tau)\,[\tau] \to \bigoplus_{\gamma\in\Delta_0^0}\bigoplus_{\tau\in N(\gamma)} R\,[\tau|\gamma]$$

such that $\delta([\tau]) = [\tau|\gamma] - [\tau|\gamma']$ for $\tau = (\gamma,\gamma') \in \Delta_1^0$, and the map

$$\varphi : \bigoplus_{\gamma\in\Delta_0^0}\bigoplus_{\tau\in N(\gamma)} R\,[\tau|\gamma] \to \bigoplus_{\gamma\in\Delta_0^0} R\,[\gamma]$$

defined by

$$\varphi([\tau|\gamma]) = \begin{cases} [\gamma] & \text{if } \gamma \in \Delta_0^0, \\ 0 & \text{if } \gamma \notin \Delta_0^0. \end{cases}$$

Then, we have $\partial_1 = \varphi \circ \delta$. We consider now the map

$$\pi : \bigoplus_{\gamma\in\Delta_0^0}\bigoplus_{\tau\in N(\gamma)} R\,[\tau|\gamma] \to \bigoplus_{\gamma\in\Delta_0^0}\bigoplus_{\tau\in N(\gamma)} R\,[\tau|\gamma]$$

such that $\pi([\tau|\gamma]) = 0$ if γ is the end point of biggest index of τ, and $\pi([\tau|\gamma]) = [\tau|\gamma]$ otherwise. Denote by $\tilde{\partial}_1 = \varphi \circ \pi \circ \delta$.

For every $\gamma \in \Delta_0^0$, let us define $\tilde{N}(\gamma)$ as the set of interior edges τ connecting γ to another vertex which is not of bigger index. Let us consider the ideal

$$\tilde{\mathcal{J}}(\gamma) = \sum_{\tau\in\tilde{N}(\gamma)} R\,\ell_\tau^{r+1} \subseteq \mathcal{J}(\gamma).$$

By construction, we have

$$\operatorname{Im}\tilde{\partial}_1 = \bigoplus_{\gamma\in\Delta_0^0} \tilde{\mathcal{J}}(\gamma)\,[\gamma]$$

and $\dim(\operatorname{Im}\partial_1)_k \geq \dim(\operatorname{Im}\tilde{\partial}_1)_k$.

Since for each edge $\tau \in \Delta_1^0$ its correspondent linear form ℓ_τ appears among the generators of the ideal $\tilde{\mathcal{J}}(\gamma)$ for (only) one interior vertex, then

$$\dim\operatorname{Im}\tilde{\partial}_1 = \sum_{i=1}^{f_0^0}\left[\tilde{t}_i\binom{k+2-(r+1)}{2} - \tilde{b}_i\binom{k+2-\tilde{\Omega}_i}{2} - \tilde{a}_i\binom{k+2-(\tilde{\Omega}_i+1)}{2}\right]$$

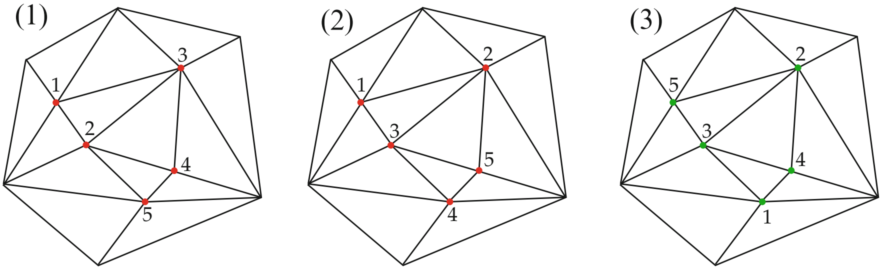


Fig. 10.1 Effect of the numbering on the upper bound

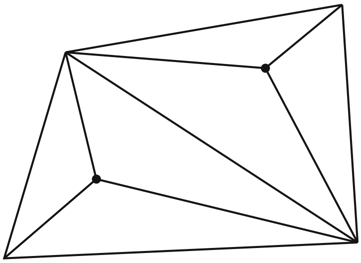


Fig. 10.2 Triangulation with two no connected interior vertices

with $\tilde{t}_i = |\tilde{N}(\gamma_i)|$ and $\tilde{\Omega}_i$, $\tilde{a}_i$, $\tilde{b}_i$ as defined in (10.11). This gives a lower bound on $\dim \operatorname{Im} \partial_1$ and proves the theorem.

Example 10.1 Effect of the ordering of the vertices on the upper bound.

Let Δ be the triangulated polygon in Fig. 10.1, and consider three different numberings of the interior vertices as shown in (1)–(3).

By Theorem 10.1, $\dim C_2^1(\Delta) \geq 9$. By Theorem 10.2 for the numbering (1), the upper bound on $\dim C_2^1(\Delta)$ is 11, for the numbering (2) the upper bound is 10, and in fact there is a numbering that give us 9 as upper bound, namely the one shown in (3). Thus, we can compute the exact dimension by combining these two bounds, $\dim C_2^1(\Delta) = 9$.

Example 10.2 Let us consider the triangulation in Fig. 10.2. It is easy to see that the lower bound on $\dim C_k^r(\Delta)$ always equals the upper bound, for any k and r.

Thus, the dimension of the spline space is given by the formula

$$\dim C_k^r(\Delta) = \binom{k+2}{2} + \binom{k+1-r}{2} + 2\binom{k-1}{2}.$$

10.4 Dimension of Trivariate Splines

Let Δ be a connected, finite 3-dimensional simplicial complex, supported on $|\Delta| \subset \mathbb{R}^3$, such that $|\Delta|$ is homotopy equivalent to a 3-dimensional ball.

In this setting the formula in (10.2) takes the form

$$\dim C_k^r(\Delta) = \sum_{i=0}^{3}(-1)^i \sum_{\beta\in\Delta_{3-i}^0} \dim \mathcal{R}/\mathcal{J}(\beta)_k + \dim H_1(\mathcal{J})_k - \dim H_0(\mathcal{J})_k \tag{10.14}$$

and by the equalities (10.6) and (10.7)

$$= \dim R_k + \dim \bigoplus_{\sigma\in\Delta_2} \mathcal{J}(\sigma)_k - \dim \operatorname{Im}(\partial_2)_k, \tag{10.15}$$

where ∂_2 is the corresponding map in the chain complex $\mathcal{J}$:

$$0 \longrightarrow \bigoplus_{\sigma\in\Delta_2^0} \mathcal{J}(\sigma) \xrightarrow{\partial_2} \bigoplus_{\tau\in\Delta_1^0} \mathcal{J}(\tau) \xrightarrow{\partial_1} \bigoplus_{\gamma\in\Delta_0^0} \mathcal{J}(\gamma) \longrightarrow 0 \tag{10.16}$$

The ring R is in this case the polynomial ring in four variables $R = \mathbb{R}[x_1, x_2, x_3, x_4]$, corresponding to the homogenization $\hat{\Delta}$ of Δ.

From (10.8) and (10.9), we have explicit formulas for the first two terms in (10.15), and thus, in order to find an upper bound on $\dim C_k^r(\Delta)$ we need to find a lower bound on the dimension of Im (∂_2) in degree k. We proceed in an analogous way as in the proof of Theorem 10.2 in the previous section.

To find an upper bound, let us consider a numbering on the interior edges τ_i of Δ, say $\tau_1, \ldots, \tau_{f_1^0}$. For each $i = 1, \ldots, f_1^0$, let s_i be the number of different linear forms defining the hyperplanes incident to τ_i, and define $\tilde{s}_i$ as the number of different linear forms defining the hyperplanes incident to τ_i, which correspond to triangles whose other two edges are either on $\partial\Delta$, or have index smaller than i. See Example 10.3.

Example 10.3 Numbering on the interior edges.

Let Δ be the Clough–Tocher split consisting of a tetrahedron which has been split about an interior point into four subtetrahedra, see Fig. 10.3.

Let us consider the numbering on the edges as in the figure. In this case, three different planes meet at each interior edges of the partition, then $s_i = 3$ for $i = 1, \ldots, 4$. On the other hand, following the counting and the definition above, $\widetilde{s_1} = 0$, $\widetilde{s_2} = 1$, $\widetilde{s_3} = 2$, and $\widetilde{s_4} = 3$.

For each $\tau_i \in \Delta_1^0$, denote by $\tilde{\mathcal{J}}(\tau_i)$ the ideal generated by the $r + 1$ powers of the linear forms defining the $\tilde{s}_i$ hyperplanes.

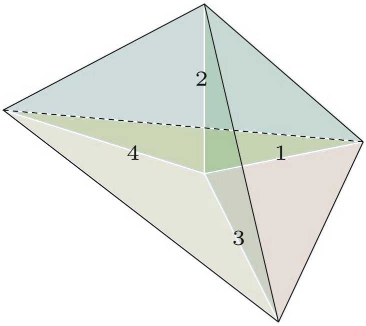


Fig. 10.3 Clough–Tocher split

Let us notice that by construction, for the edges $\tau_i \in \Delta_i^0$ of each triangle σ in Δ_2^0, the linear form ℓ_σ is among the generators of $\tilde{\mathcal{J}}(\tau_i)$ for only one τ_i, namely that with highest index.

Theorem 10.3 *The dimension of $C_k^r(\Delta)$ is bounded above by*

$$\dim C_k^r(\Delta) \leq \binom{k+3}{3} + f_2^0\binom{k+2-r}{3}$$
$$-\sum_{i=1}^{f_1^0}\left[\tilde{s}_i\binom{k+2-r}{3} - \tilde{b}_i\binom{k+3-\tilde{\Omega}_i}{3} - \tilde{a}_i\binom{k+2-\tilde{\Omega}_i}{3}\right]$$

with $\tilde{s}_i$ as we have defined above and

$$\tilde{\Omega}_i = \left\lfloor \frac{\tilde{s}_i\, r}{\tilde{s}_i - 1} \right\rfloor + 1, \quad \tilde{a}_i = \tilde{s}_i\,(r+1) + (1-\tilde{s}_i)\,\tilde{\Omega}_i, \quad \tilde{b}_i = \tilde{s}_i - 1 - \tilde{a}_i.$$

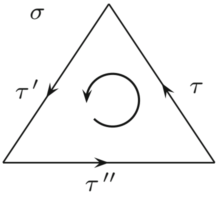
If $\tilde{s}_i = 1$ or 0 then $\tilde{a}_i = \tilde{b}_i = \tilde{\Omega}_i = 0$.

Proof We define the maps δ, φ and π by R-linear extensions as follows. Let δ be the map

$$\delta : \bigoplus_{\sigma=(\tau,\tau',\tau'')} \mathcal{J}(\sigma)[\sigma] \to \bigoplus_{\tau_i\in\Delta_1^0}\bigoplus_{\sigma\in N(\tau_i)} \mathcal{R}[\sigma|\tau_i]$$

where, for each i, $N(\tau_i)$ denotes the set of triangles that contain the edge τ_i. The map δ is induced by the boundary map ∂_2. Thus, $\delta([\sigma]) = [\sigma|\tau] - [\sigma|\tau'] + [\sigma|\tau'']$ for $\sigma = (\tau,\tau',\tau'') \in \Delta_2^0$, see Fig. 10.4.

Fig. 10.4 Orientation of a triangle $\sigma \in \Delta_2^0$



$$\partial_2 [\sigma] = [\tau] - [\tau'] + [\tau'']$$

Let

$$\varphi : \bigoplus_{\tau_i \in \Delta_1^0} \bigoplus_{\sigma \in N(\tau_i)} \boldsymbol{\mathcal{R}}[\sigma|\tau_i] \to \bigoplus_{\tau_i \in \Delta_1^0} \boldsymbol{\mathcal{R}}[\tau_i]$$

with

$$\varphi([\sigma|\tau_i]) = \begin{cases} [\tau_i] & \text{if } \tau_i \in \Delta_1^0 \\ 0 & \text{if } \tau_i \notin \Delta_1^0. \end{cases}$$

Then $\partial_2 = \varphi \circ \delta$. We consider the map

$$\pi : \bigoplus_{\tau_i \in \Delta_1^0} \bigoplus_{\sigma \in N(\tau_i)} \boldsymbol{\mathcal{R}}[\sigma|\tau_i] \to \bigoplus_{\tau_i \in \Delta_1^0} \bigoplus_{\sigma \in N(\tau_i)} \boldsymbol{\mathcal{R}}[\sigma|\tau_i]$$

defined by $\pi([\sigma|\tau]) = [\sigma|\tau]$, $\pi([\sigma|\tau']) = \pi([\sigma|\tau'']) = 0$ if $\sigma = (\tau, \tau', \tau'')$ and either τ' or τ'' are on the boundary of Δ or the index assigned to them is bigger than the assigned to τ. Denote by $\tilde{\partial}_2 = \varphi \circ \pi \circ \delta$.

For $\tau_i \in \Delta_1^0$, let $\tilde{N}(\tau_i)$ be the set of triangles $\sigma \in \Delta_2^0$ that contain τ as an edge and whose other two edges do not have bigger index than the index of τ. Then we have,

$$\tilde{\mathcal{J}}(\tau_i) = \sum_{\sigma \in \tilde{N}(\tau_i)} R\ell_\sigma^{r+1} \subseteq \mathcal{J}(\tau_i).$$

By construction,

$$\operatorname{Im} \tilde{\partial}_2 = \bigoplus_{\tau_i \in \Delta_1^0} \tilde{\mathcal{J}}(\tau_i)[\tau_i]$$

and $\dim \operatorname{Im} (\partial_2)_k \geq \dim \operatorname{Im} (\tilde{\partial}_2)_k$. Thus, from the formula for $\dim C_k^r(\Delta)$ given in (10.15), it follows

$$\dim C_k^r(\Delta) \leq \dim \boldsymbol{\mathcal{R}}_k + \dim \bigoplus_{\sigma \in \Delta_2} \mathcal{J}(\sigma)_k - \dim \operatorname{Im} (\tilde{\partial}_2).$$

By an affine change of coordinates we may assume the edge τ_i to be along one of the coordinate edges, and thus the linear forms in $\tilde{\mathcal{J}}(\tau_i)$ only involve two variables.

Then $\dim \operatorname{Im}(\tilde{\partial}_2)_k$ is given by (10.12) with $d = 3$, $\tilde{s}_i = |\tilde{N}(\tau_i)|$, and $\tilde{\Omega}_i$, $\tilde{a}_i$ and $\tilde{b}_i$ as given by the formulas (10.11) with $\tilde{s}_i$ instead of t_i. This together with (10.8) and (10.9) proves the theorem.

Our aim now is to provide a lower bound on the dimension of $C^r_k(\Delta)$ to complement the upper bound in Theorem 10.3.

Let us take zero as a lower bound for $\dim H_1(\mathcal{J})_k$, then from (10.14), for any $k \geq 0$:

$$\dim C^r_k(\Delta) \geq \dim R_k + \sum_{i=1}^{2}(-1)^i \dim \bigoplus_{\beta\in\Delta^0_{3-i}} \mathcal{J}(\beta)_k + \dim \operatorname{Im}(\partial_1)_k \qquad (10.17)$$

Since we have explicit formulas for all the terms in (10.17) except for the last one. We can establish a lower bound on $\dim C^r_k(\Delta)$ by finding a lower bound on $\dim \operatorname{Im}(\partial_1)$.

We proceed analogously as before. This time we give an ordering to the vertices in the interior of Δ. For each vertex γ_i, we denote by $M(\gamma_i)$ the set of edges τ in Δ^0_1 that contain the vertex γ_i, and by $\tilde{M}(\gamma_i)$ be the set of interior edges connecting γ_i to one of the first $i-1$ vertices in the list, or to a vertex in the boundary.

For each $\gamma_i \in \Delta^0_0$, let t_i be defined as before, the number of generators of $\mathcal{J}(\gamma_i)$. Define the ideal $\tilde{\mathcal{J}}(\gamma_i)$ as

$$\tilde{\mathcal{J}}(\gamma_i) = \langle \ell_\sigma^{r+1}\rangle_{\sigma\ni\tau} \quad \text{for } \tau \in \tilde{M}(\gamma_i),$$

and let $\tilde{t}_i$ be the number of generators of $\tilde{\mathcal{J}}(\gamma_i)$. An analogous argument to the one we used in the proofs above leads to the following lemma.

Lemma 10.1 *For a three dimensional simplicial complex Δ, the dimension of the spline space $C^r_k(\Delta)$ is bounded by*

$$\dim C^r_k(\Delta) \geq \dim R_k + \sum_{i=1}^{2}(-1)^i \dim \bigoplus_{\beta\in\Delta^0_{3-i}} \mathcal{J}(\beta)_k + \dim \bigoplus_{i=1}^{f^0_0} \tilde{\mathcal{J}}(\gamma_i)_k. \qquad (10.18)$$

Proof Follows from (10.17), and the argument as before.

We have explicit formulas for the terms in (10.18), except for $\dim \bigoplus_{i=1}^{f^0_0} \tilde{\mathcal{J}}(\gamma_i)_k$. This term involves ideals generated by powers of linear forms in three variables, which correspond to equations of planes going through a common point.

As we mentioned above, there is a formula $F(t, r+1, n)$ conjectured by Fröberg [12] concerning the dimension of ideals generated by a generic set of forms (not necessarily powers of linear forms) in a polynomial ring in n variables. This conjecture, in particular, was proved for the case $n = 3$ of forms in three variables [3]. But since the conjectured dimension is not true in general when the generators are powers of linear forms [21], such formula only gives us a lower bound on the dimension of $\mathcal{R}/\mathcal{J}(\gamma)_k$ for every vertex γ, namely

$$\dim \mathcal{R}/\mathcal{J}(\gamma)_k \geq \sum_{j=0}^{k} F(t, r+1, 3)_j, \tag{10.19}$$

where t is the number of linear forms that corresponds to the different planes that contain the vertex γ in Δ. Equality holds in (10.19) when $t \leq 3$ for each vertex $\gamma \in \Delta_0^0$ [15].

The formula $F(t, r+1, n)$ associated to the Hilbert function of an ideal generated by t forms of degree $r+1$ in a polynomial ring of $n = 3$ variables over $\mathbb{R}$ (or any field of characteristic zero) is be defined as follows, for $j \geq 0$

$$F(t, r+1, 3)_j = \begin{cases} F'(t, r+1, 3)_j, & \text{if } F'(t, r+1, 3)_u > 0 \text{ for all } u \leq j, \\ 0 & \text{otherwise;} \end{cases} \tag{10.20}$$

where $F'(t, r+1, 3)_j$ is given by

$$F'(t, r+1, 3)_j = \dim R_j + \sum_{v=1}^{3} (-1)^v \dim R_{j-(r+1)v} \binom{t}{v}.$$

It is a particular case of [15, Theorem 1.6].

Theorem 10.4 *The dimension* $\dim C_k^r(\Delta)$ *is bounded below by*

$$\begin{aligned} \dim C_k^r(\Delta) \geq \binom{k+3}{3} + \Bigg[f_2^0 \binom{k+2-r}{3} & \\ - \sum_{i=1}^{f_1^0} \left[s_i \binom{k+2-r}{3} - b_i \binom{k+3-\Omega_i}{3} - a_i \binom{k+2-\Omega_i}{3} \right] & \\ + f_0^0 \binom{k+3}{3} - \sum_{i=1}^{f_0^0} \left(\sum_{j=0}^{k} F(\zeta_i, r+1, 3)_j \right) \Bigg]_+ & \end{aligned} \tag{10.21}$$

with $\zeta_i = \min(3, \tilde{t}_i)$*,* s_i *as defined above, and*

$$\Omega_i = \left\lfloor \frac{s_i\, r}{s_i - 1} \right\rfloor + 1, \quad a_i = s_i\,(r+1) + (1 - s_i)\,\Omega_i, \quad b_i = s_i - 1 - a_i.$$

Proof It is clear from Lemma 10.1 and the previous remarks. Since the dimension of the spline space is at least the number of polynomials in tree variables of degree less than or equal to k, then we take the positive part of the additional terms in (10.21). See [21] for the detailed proof.

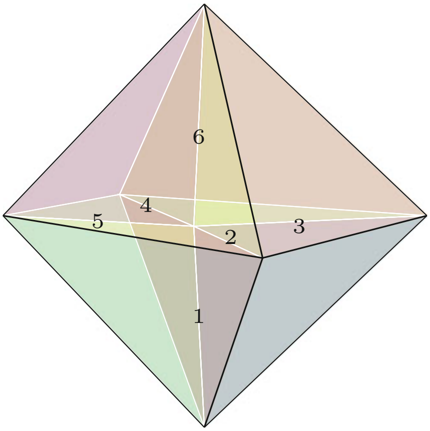


Fig. 10.5 Regular octahedron

Remark 10.1 The lower bound on $C^r_k(\Delta)$ in the previous theorem can be improved if the linear forms defining the ideals $\mathcal{J}(\gamma_i)$ are generic, or if the Hilbert function of ideals generated by powers of $\tilde{t}_i \geq 4$ linear forms in three variables is known, in which case one might avoid the step of taking $\zeta_i = \min(3, \tilde{t}_i)$. In consequence, the results in algebraic geometry about the Hilbert function of ideals of powers of linear forms and the related ideal of fat points would significantly improve results concerning the dimension of spline spaces.

For the central configurations that we will consider in this section, it is easy to see that $H_0(\mathcal{J})$ is always zero, see [21] for more details.

Example 10.4 Let Δ be a octahedron subdivided into eight tetrahedra by placing a symmetric central vertex, see Fig. 10.5.

Computations show that $H_1(\mathcal{J})$ is zero for all non-generic octahedra [23]. Since in this partition, there are exactly three different planes through the central vertex, then the Fröberg sequence gives us an explicit formula for the dimension of the ideal associated to the (unique) interior vertex. Hence the dimension $\dim C^r_k(\Delta)$ can be directly computed using (10.14) as follows,

$$\begin{aligned}\dim C^r_k(\Delta) = &\binom{k+3}{3} + 12\binom{k+3-(r+1)}{3}\\ &-\sum_{i=1}^{6}\left[2\binom{k+3-(r+1)}{3} - \binom{k+3-(2r+2)}{3}\right]\\ &+\binom{k+3}{3} - \sum_{j=0}^{k} F(3, r+1, 3)_j .\end{aligned}$$

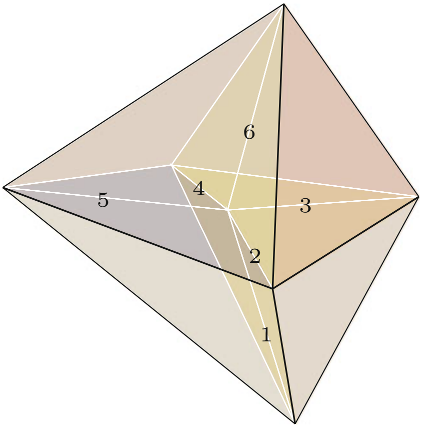


Fig. 10.6 Generic octahedron

From the definition of Fröberg's sequence (10.20),

$$F(3,r+1,3)_j = \binom{j+2}{2} - 3\binom{j+1-r}{2} + 3\binom{j-2r}{2} - \binom{j-3r-1}{2}.$$

It is easy to check that $F(3,r+1,3)_j > 0$ for every $0 \leq j < 3r+3$, and equal to zero otherwise. Hence, we can write

$$\sum_{j=0}^{k} F(3,r+1,3)_j = \binom{k+3}{3} - 3\binom{k+2-r}{3} + 3\binom{k-2r+1}{3} - \binom{k-3r}{3} \tag{10.22}$$

and thus, the formula for the dimension of the spline space on the regular octahedron in Fig. 10.5 is given by the expression

$$\dim C_k^r(\Delta) = \binom{k+3}{3} + 3\binom{k+2-r}{3} + 3\binom{k+1-2r}{3} + \binom{k-3r}{3}.$$

Example 10.5 Let us consider the generic case of an octahedron subdivided into tetrahedra, where no set of four vertices of the octahedron is coplanar, Fig. 10.6. As we mentioned above, we have $H_0(\mathcal{J})$ equal to zero. But in contrast to the regular case, $H_1(\mathcal{J})$ is equal to zero when $r = 1$ but not for any other value of r [23].

For this partition Δ, we have $t = 12$ different planes corresponding to the triangles meeting at the central vertex. Then $\zeta = \min(3,t) = 3$, and using the formula (10.22) from the previous example for the sum of the $F(3, r+1, 3)_j$ for $r = 1$, Theorem 10.4 gives us the following lower bound

$$\begin{aligned}\dim C^1_k(\Delta) \geq \binom{k+3}{3} + \Bigg[12\binom{k+1}{3} - 6\bigg[3\binom{k+1}{3} - 2\binom{k}{3}\bigg]\\ + \binom{k+3}{3} - \sum_{j=0}^{k} F(3,2,3)_j\Bigg]_+\\ = \binom{k+3}{3} + \left[-3\binom{k+1}{3} + 12\binom{k}{3} - 3\binom{k-1}{3} + \binom{k-3}{3}\right]_+.\end{aligned}$$

In order to find an upper bound, we apply Theorem 10.3 for some ordering on the interior edges of the partition. For instance, with the numbering on the edges as in Fig. 10.6, we have $\tilde{s}_1 = 0$, $\tilde{s}_2 = 1$, $\tilde{s}_3 = \tilde{s}_4 = 2$, $\tilde{s}_5 = 3$, and $\tilde{s}_6 = 4$, and so for any degree k:

$$\dim C^1_k(\Delta) \leq \binom{k+3}{3} + \binom{k+1}{3} + 4\binom{k}{3} + 2\binom{k-1}{3}.$$

Example 10.6 Let Δ be the Clough–Tocher split consisting of a tetrahedron which has been split about an interior point into four subtetrahedra, Fig. 10.3.

We consider $r = 1$ and $r = 2$. In these two cases the homology module $H_1(\mathcal{J})$ is zero.

(i) For $r = 1$, as in the previous example, we have

$$\sum_{j=0}^{k} F(3,2,3)_j = \binom{k+3}{3} - 3\binom{k+1}{3} + 3\binom{k-1}{3} - \binom{k-3}{3}$$

Then, the lower bound on the spline space proved in Theorem 10.4 is given by

$$\dim C^1_k(\Delta) \geq \binom{k+3}{3} + \left[-3\binom{k+1}{3} + 8\binom{k}{3} - 3\binom{k-1}{3} + \binom{k-3}{3}\right]_+.$$

The upper bound we obtained in this example, by applying Theorem 10.3 with the numbering of the edges as in Fig. 10.3 is the following:

$$\dim C^1_k(\Delta) \leq \binom{k+3}{3} + \binom{k-1}{3} + 2\binom{k}{3}.$$

Since for $r = 1$ the homology module $H_1(\mathcal{J}) = 0$, then we can apply (10.14) and the bound (10.19) with $t = 6$ in the sequence (10.20) for the (unique) interior vertex in Δ. This leads to the following upper bound

$$\dim C_k^1(\Delta) \leq \begin{cases} 1 & \text{for } k = 0 \\ 2\binom{k+3}{3} - 6\binom{k+1}{3} + 8\binom{k}{3} - 4 & \text{for } k \geq 1. \end{cases} \tag{10.23}$$

The formula (10.23) coincides with the generic dimension formula computed in [2] for this partition Δ. Although the formula in [2] holds only for $k \geq 8$ (and $r = 1$), it in turn coincides with the lower bound formula proved in [1] in every degree $k \geq 0$. In fact, in general, the dimension of the spline space of any nongeneric decomposition is always greater than or equal to the generic dimension, it is the smallest dimension encountered as one moves the vertices of the complex. Thus, since the lower bound formula proved in [1] coincides with the upper bound we proved above (10.23), we deduce the following result: *the exact dimension of the C^1 spline space over the Clough–Tocher split is*

$$\dim C_k^1(\Delta) = \begin{cases} 1 & \textit{for } k = 0 \\ 2\binom{k+3}{3} - 6\binom{k+1}{3} + 8\binom{k}{3} - 4 & \textit{for } k \geq 1. \end{cases}$$

(ii) Let us consider the case $r = 2$.

A lower bound is given by the formula

$$\dim C_k^2(\Delta) \geq \binom{k+3}{3} + \left[-3\binom{k}{3} + 4\binom{k-1}{3} + 4\binom{k-2}{3} - 3\binom{k-3}{3} + \binom{k-6}{3}\right]_+$$

Using that $H_1(\mathcal{J}) = 0$, and (10.14), (10.19) and (10.20) as before, the following is an upper bound for $k \geq 3$:

$$\dim C_k^2(\Delta) \leq 2\binom{k+3}{3} - 6\binom{k}{3} + 4\binom{k-1}{3} + 4\binom{k-2}{3} - 14$$

The values of the previous bounds on $\dim C_k^2(\Delta)$ for $k \leq 9$ are given in the following table. The first row shows the values obtained using the lower bound formula from [1].

k	1	2	3	4	5	6	7	8	9
Lower bound [1]	4	10	20	35	56	84	120	179	261
Lower bound	4	10	20	35	56	84	123	187	282
Upper bound	4	10	20	36	58	90	136	200	286

Remark 10.2 The examples above illustrate the improvement that our lower and upper bounds provide with respect to previous results in the literature. Furthermore, as we showed in the last example, the formulas we presented here might be combined with results obtained by using different techniques leading thus to sharper bounds, and in many cases to the exact dimension of the space.

Remark 10.3 The approaches we use in this work differ from the ones used before to find bounds on the dimension of a spline space defined on a simplicial complex, see [16] and the references therein. The results and examples we presented give an insight into ways of improving the bounds and finding the exact dimension formula under certain conditions. In [21] and [31], the reader can find a more extended discussion on the relationship between splines and fat points, and the connection of that theory with the Weak Lefschetz Property [14, 18], Hilbert series of ideals of powers of generic linear forms, and Fröberg's conjecture and its most recent versions [8].

References

1. P. Alfeld, L.L. Schumaker, Bounds on the dimensions of trivariate spline spaces. Adv. Comput. Math. **29**(4), 315–335 (2008)
2. P. Alfeld, L.L. Schumaker, W. Whiteley, The generic dimension of the space of C^1 splines of degree $d \geq 8$ on tetrahedral decompositions. SIAM J. Numer. Anal. **30**(3), 889–920 (1993)
3. D.J. Anick, Thin algebras of embedding dimension three. J. Algebra **100**(1), 235–259 (1986)
4. L.J. Billera, Homology of smooth splines: generic triangulations and a conjecture of Strang. Trans. Am. Math. Soc. **310**(1), 325–340 (1988)
5. L.J. Billera, The algebra of continuous piecewise polynomials. Adv. Math. **76**(2), 170–183 (1989)
6. L.J. Billera, L.L. Rose, A dimension series for multivariate splines. Discret Comput. Geom. **6**(2), 107–128 (1991)
7. L.J. Billera, L.L. Rose, Modules of piecewise polynomials and their freeness. Math. Z. **209**(4), 485–497 (1992)
8. C. Ciliberto, Geometric aspects of polynomial interpolation in more variables and of Waring's problem, in *European Congress of Mathematics, Vol. I (Barcelona, 2000)*, ed. by C. Casacuberta, R.M. Miró-Roig , J. Verdera, S. Xambó-Descamps. Progress in Mathematics, vol. 201 (Birkhäuser, Basel, 2001), pp. 289–316
9. J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA* (Wiley, Chichester/Hoboken, 2009)
10. G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*: A practical Guide. Computer Science and Scientific Computing, 4th ed. (Academic Press, Boston, 1997)
11. G. Fix, G. Strang, *An Analysis of the Finite Element Method*, 2nd edn. (Wellesley-Cambridge, Wellesley, 2008)
12. R. Fröberg, An inequality for Hilbert series of graded algebras. Math. Scand. **56**(2), 117–144 (1985)
13. A.V. Geramita, H. Schenck, Fat points, inverse systems, and piecewise polynomial functions. J. Algebra **204**(1), 116–128 (1998)
14. B. Harbourne, H. Schenck, A. Seceleanu. Inverse systems, Gelfand-Tsetlin patterns the weak Lefschetz property. J. Lond. Math. Soc. (2) **84**(3), 712–730 (2011)

15. A. Iarrobino, Inverse system of a symbolic power. III. Thin algebras and fat points. Compositio Math. **108**(3), 319–356 (1997)
16. M-J. Lai, L.L. Schumaker, *Spline Functions on Triangulations*. Encyclopedia of Mathematics and its Applications, vol. 110 (Cambridge University Press, Cambridge, 2007)
17. W. Lau, A lower bound for the dimension of trivariate spline spaces. Constr. Approx. **23**(1), 23–31 (2006)
18. J. Migliore, R. Miró-Roig, U. Nagel, On the weak Lefschetz property for powers of linear forms. Algebra Number Theory **6**(3), 487–526 (2012)
19. J. Morgan, R. Scott, A nodal basis for C^1 piecewise polynomials of degree $n \geq 5$. Math. Comput. **29**, 736–740 (1975)
20. B. Mourrain, N. Villamizar, Homological techniques for the analysis of the dimension of triangular spline spaces. J. Symb. Comput. **50**, 564–577 (2013)
21. B. Mourrain, N. Villamizar, Dimension of spline spaces on tetrahedral partitions: a homological cal approach. Math. Comput. Sci. **8**(2), 157–174.
22. J.R. Munkres, *Elements of Algebraic Topology* (Addison-Wesley, Menlo Park, 1984)
23. H. Schenck, A spectral sequence for splines. Adv. Appl. Math. **19**(2), 183–199 (1997)
24. H. Schenck, M. Stillman, Local cohomology of bivariate splines. J. Pure Appl. Algebra **117/118**, 535–548 (1997). Algorthms for Algebra (Eindhoven, 1996)
25. H. Schenck, M. Stillman, A family of ideals of minimal regularity and the Hilbert series of $C^r(\hat{\Delta})$. Adv. Appl. Math. **19**(2), 169–182 (1997)
26. I.J. Schoenberg, Contributions to the problem of approximation of equidistant data by analytic functions. Part A. On the problem of smoothing or graduation. A first class of analytic approximation formulae. Part B. On the problem of osculatory interpolation. A second class of analytic approximation formulae. Quart. Appl. Math. **4**, 45–99, 112–141 (1946)
27. L.L. Schumaker, Bounds on the dimension of spaces of multivariate piecewise polynomials. Rocky Mt. J. Math. **14**(1), 251–264 (1984). Surfaces (Stanford, California, 1982)
28. E.H. Spanier, *Algebraic Topology* (McGraw-Hill, New York, 1966)
29. G. Strang, Piecewise polynomials and the finite element method. Bull. Am. Math. Soc. **79**, 1128–1137 (1973)
30. G. Strang, The dimension of piecewise polynomial spaces, and one-sided approximation, in *Conference on the Numerical Solution of Differential Equations*, University Dundee, Dundee, 1973, ed. by G.A. Watson. Lecture Notes in Mathematics, vol. 363 (Springer, Berlin, 1974), pp. 144–152
31. N. Villamizar, Algebraic geometry for splines, Doctoral dissertation, University of Oslo, 2012. http://urn.nb.no/URN:NBN:no-38634

Chapter 11
Polynomial Interpolation Problems in Projective Spaces and Products of Projective Lines

Elisa Postinghel

11.1 Introduction

The classical polynomial interpolation theory of functions arises in numerical analysis and statistics and is based on the fact that a univariate polynomial of fixed positive degree d is uniquely determined by its values at $d + 1$ distinct points on the affine line, due to the non-singularity of the Vandermonde matrix. One can ask not only for the value of the polynomial but also of its derivatives, up to some given order, at a finite number of distinct points. The corresponding homogeneous problem, consisting of asking that the assigned distinct points are multiple roots of the polynomial, is a full-rank linear problem in the vector space of polynomials of degree d and has non-trivial solutions provided that the sum of all multiplicities equals at most the degree.

The first generalization is the multivariate polynomial interpolation problem: one can ask for the dimension of the vector space of homogeneous (or multi-homogeneous) polynomials of fixed degree (resp. multi-degree) that vanish, together with their partial derivatives up to fixed order, at a finite number of assigned distinct points. This interpolation problem does not depend on the choice of coordinates and it is referred to as *Hermite interpolation*. Unlike the one-variable case, where the points are only required to be distinct, in the case of more variables the dimension of such a vector space depends on the position of the points; it reaches its minimum value when the points are in general position. In spite of its easy formulation, surprisingly enough there is no complete answer to this problem so far.

E. Postinghel (✉)
Department of Mathematics, KU Leuven, Celestijnenlaan 200b, 3001 Heverlee, Belgium
e-mail: elisa.postinghel@wis.kuleuven.be

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_11

Hermite interpolation problems for homogeneous or multi-homogeneous polynomials lend themselves to being stated in the setting of linear systems and points in projective spaces or products of projective spaces. The main question we address is the so called *dimensionality problem* for linear systems which is: given a linear system of hypersurfaces of a variety X and a general set of points of X, what is the dimension of the linear subsystem formed by those elements having at least the assigned multiplicity at the given points? In Sect. 11.2 we give a more precise statement of the problem.

In Sects. 11.3 and 11.4 we consider the case where $X = \mathbb{P}^n$. In Sect. 11.3 we discuss the state of the art introducing the main results and conjectures.

In Sect. 11.4 we describe our main contribution [8] in this direction and explain the techniques used. We also give a brief account on interesting applications of our results in the commutative algebra setting of Fröberg-Iarrobino Conjectures on the Hilbert series of the ideals generated by powers of general linear forms.

In Sects. 11.5 and 11.6 we consider the case where $X = \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$. In Sect. 11.5 we relate it to the study of secant varieties of Segre-Veronese varieties, the Waring problems and the rank of partially symmetric tensors.

In Sect. 11.6 we present our contribution [26] which is the complete classification of linear systems with double points on the product $\mathbb{P}^1 \times \cdots \times \mathbb{P}^1$. Our approach to the problem employs toric degenerations of Segre-Veronese varieties (Sect. 11.6.2).

11.2 Linear Systems with Multiple Base Points

Let X be a smooth, irreducible, complex projective variety of dimension n. Let $\mathcal{L}_X$ denote the linear system of hyperplane sections of X. Fix $p_1, \dots, p_s$ distinct points on X in general position, i.e. in a Zariski open set (which means outside the zero locus of a polynomial), and fix $m_1, \dots, m_s$ positive integers. We will denote by $\mathcal{L} = \mathcal{L}_X(m_1, \dots, m_s)$ the linear subsystem of $\mathcal{L}_X$ formed by all elements in $\mathcal{L}_X$ having multiplicity at least m_i at p_i, $i = 1, \dots, s$.

If $x_0, \dots, x_n$ are local coordinates centred at p_i and $f(x_0, \dots, x_n) = 0$ is the equation of a divisor of $\mathcal{L}_X$, then such a divisor belongs to $\mathcal{L}_X(m_i)$ if all monomials of degree at most $m_i - 1$ appearing in the Taylor expansion of $f(x_0, \dots, x_n)$ vanish at p_i, for $i = 1, \dots, s$. This imposes $\binom{n+m_i-1}{n}$ linear conditions to the coefficients of $f(x_0, \dots, x_n)$. Accordingly, the *virtual dimension* of $\mathcal{L}_X(m_1, \dots, m_s)$ is defined as

$$\operatorname{vdim}(\mathcal{L}) := \dim(\mathcal{L}_X) - \sum_{i=1}^{s} \binom{n + m_i - 1}{n} - 1,$$

and the *expected dimension* as

$$\operatorname{edim}(\mathcal{L}) := \max\left(\operatorname{vdim}(\mathcal{L}), -1\right),$$

using the convention that the empty set has dimension -1. The dimension of $\mathcal{L}$ is upper semi-continuous in the position of the points in X; it achieves its minimum value when they are in general position. Whereas the inequality $\dim(\mathcal{L}) \geq \operatorname{edim}(\mathcal{L})$ is always satisfied, the actual dimension of $\mathcal{L}$ is strictly greater than the expected one if the conditions imposed by the assigned points are not linearly independent: in that case we say that $\mathcal{L}$ is *special*. Otherwise, if the actual and the expected dimension coincide, we say that $\mathcal{L}$ is *non-special.*

One naturally expects that most systems are non-special, at least for low multiplicities. This immediately turns out to be false. Consider for instance the linear system of plane conics and impose two distinct double points p_1 and p_2. Since each of them imposes three conditions to a curve in the projective plane, with a count of parameters one expects that the system is empty. But if $l(x_0, x_1, x_2) = 0$ is the linear equation defining the line through p_1 and p_2, then $l(x_0, x_1, x_2)^2 = 0$ is a conic which is singular at p_1 and p_2, therefore such a linear system does not have the expected dimension.

The so called *dimensionality problem* for linear systems with base points in general position is: classify all special systems.

One may also ask more refined questions about linear systems such as: describe their base locus, namely the locus along which all elements of the linear systems vanish. Notice that in the example of conic curves through two double points described above, the unique element of the linear system, namely $l(x_0, x_1, x_2)^2 = 0$, is singular along the whole line. The same geometric argument applies to the system of quadrics of $\mathbb{P}^3$ with two double points: all hypersurfaces in the system are singular along the line joining p_1 and p_2, therefore the line is contained with multiplicity (at least) two in the base locus of the linear system. Imposing a third base point to the system, say p_3, one is implicitly imposing that both lines spanned by p_1, p_3 and by p_2, p_3 are base lines. A similar phenomenon occurs with plane quartics with five double points. Indeed one expects to have none of them. On the contrary, as there exists a plane conic through five general points, say $f(x_0, x_1, x_2) = 0$, then the quartic $f(x_0, x_1, x_2)^2 = 0$ belongs to the system and is contained in its base locus.

In spite of their easy formulation, these problems are very hard and challenging. Very little is known, although many efforts have been made in the last century. In the next sections we will discuss previously known results and conjectures and we will present our contributions to these kinds of problems. In Sect. 11.4 a detailed description of the linear components of the base locus of a linear system in $X = \mathbb{P}^n$ is given and, the dimensionality problem for linear systems with bounded number of points or multiplicities is answered. In Sect. 11.6 a complete classification of special linear systems in the case where X is the Segre-Veronese embedding of $(\mathbb{P}^1)^n$ and $m_1 = \cdots = m_s = 2$ is established.

11.3 Prescribing Multiple Points in $\mathbb{P}^n$

The study of linear systems of hypersurfaces in complex projective spaces with finitely many assigned base points of given multiplicities is a fundamental problem in algebraic geometry, related to the Waring problem for polynomials and to the classification of defective higher secant varieties to projective varieties.

Let $\mathcal{L} = \mathcal{L}_{n,d}(m_1, \dots, m_s)$ be the linear system of hypersurfaces of degree d in $\mathbb{P}^n$ passing through a general union of s points $p_1, \dots, p_s$ with multiplicity respectively $m_1, \dots, m_s \leq d$. The *virtual dimension* of $\mathcal{L}$ is

$$\operatorname{vdim}(\mathcal{L}) = \binom{n+d}{n} - \sum_{i=1}^{s} \binom{n+m_i-1}{n} - 1$$

and the *expected dimension* of $\mathcal{L}$ is $\operatorname{edim}(\mathcal{L}) = \max(\operatorname{vdim}(\mathcal{L}), -1)$. The linear system $\mathcal{L}$ is said to be *special* if $\dim(\mathcal{L}) > \operatorname{edim}(\mathcal{L})$ and *non-special* if $\dim(\mathcal{L}) = \operatorname{edim}(\mathcal{L})$. The *speciality* of $\mathcal{L}$ is defined to be the difference $\dim(\mathcal{L}) - \operatorname{edim}(\mathcal{L})$.

Remark 11.1 A geometric interpretation of the notion of speciality for a linear system is the following. Let X be the blow-up X of $\mathbb{P}^n$ at the points p_i's and let us denote by H the pull-back of the class of a hyperplane of $\mathbb{P}^n$ and by E_i's the classes of the exceptional divisors of the points p_i's, so that the Picard group of X is $\operatorname{Pic}(X) = \langle H, E_i : i = 1, \dots, s \rangle$. The proper transform D of an element in $\mathcal{L}$ is linearly equivalent to the divisor

$$dH - \sum_{i=1}^{s} m_i E_i \in \operatorname{Pic}(X).$$

By abuse of notation we will use the same letter $\mathcal{L}$ to denote the linear system in X associated to D, when no confusion arises. The following equalities hold

- $\dim(\mathcal{L}) = h^0(X, \mathcal{O}_X(D)) - 1,$
- $\operatorname{vdim}(\mathcal{L}) = \chi(\mathcal{O}_X(D)) - 1 = h^0(X, \mathcal{O}_X(D)) - h^1(X, \mathcal{O}_X(D)) - 1,$

where h^i denotes the dimension of the i-th cohomology group and χ denotes the Euler characteristic of a sheaf. Indeed, set $D(j,k)$ to be the divisor

$$dH - \sum_{i=1}^{j-1} m_i E_i - kE_j \in \operatorname{Pic}(X),$$

with $1 \leq j \leq s$ and $1 \leq k \leq m_j$. From the long exact sequences in cohomology associated with the short exact sequences

$$0 \to D(j,k) \to D(j,k) + E_j \to (D(j,k) + E_j)|_{E_j} \to 0,$$

obtained by following the lexicographic order on the set of indices $\{(j,k) : 1 \leq j \leq s, 1 \leq k \leq m_j\}$, one obtains $h^i(X, D(j,k)) = 0$, i$\geq$ 2. Hence $\mathcal{L}$ is non-special if and only if

$$h^0(X, \mathcal{O}_X(D)) \cdot h^1(X, \mathcal{O}_X(D)) = 0.$$

Remark 11.2 The multiplicity conditions give a homogeneous ideal in the homogeneous coordinate ring of the projective space; the graded pieces of such an ideal, as we vary d, correspond to the linear systems $\mathcal{L}$. In this way determining the dimension of $\mathcal{L}$, for different values of d, is equivalent to computing the Hilbert function of the graded ideal of the points, providing an algebraic presentation of the notion of speciality for linear systems.

The problem of determining the speciality of linear systems of hypersurfaces of $\mathbb{P}^n$ with fixed degree and prescribed multiplicities at a given collection of points attracted the attention of many researchers in the last century. What is known is essentially concentrated in the case of double base points, namely $m_1 = \cdots = m_s = 2$. A complete classification of special linear systems with double base points in general position was proved by Alexander and Hirschowitz [3] by means of the so called Horace's method. Recent and simplified proofs can be found in [9] and [29]. The theorem states that a linear system in $\mathbb{P}^n$ with only double points is special, besides four exceptional cases, only if the degree is 2 and the number of points is $2 \leq s \leq n$. The case of quadrics is easily understood. Indeed on the one hand the linear system $\mathcal{L} = \mathcal{L}_{n,2}(2^s) = \mathcal{L}_{n,2}(2,\ldots,2)$ satisfies $\operatorname{vdim}(\mathcal{L}) = \binom{n+2}{2} - s(n+1) - 1$; on the other hand it is easy to see that any hypersurface in $\mathcal{L}$ is a quadric cone with vertex the linear subspace $\mathbb{P}^{s-1}$ spanned by the s points, hence $\dim(\mathcal{L}) = \binom{n-s+2}{2} - 1$, which is the dimension of the complete linear system of quadric hypersurfaces in $\mathbb{P}^{n-s}$. One can now check that $\dim(\mathcal{L}) > \operatorname{edim}(\mathcal{L})$ for $2 \leq s \leq n$. The quartics of $\mathbb{P}^2$ with five double points, that we already discussed in Sect. 11.2, are special and, for the same reason, so are the quartics of $\mathbb{P}^3$ and $\mathbb{P}^4$ with respectively 9 and 14 double points. Finally, the cubics of $\mathbb{P}^4$ with seven double points fall into the list of special cases, see for instance [29] for a detailed description of these exceptional cases.

The problem becomes more and more complicated for higher multiplicities and is open so far. It is related for $n > 2$ to the Fröberg-Iarrobino conjecture, which gives a predicted value for the Hilbert series of an ideal generated by s general d-powers of linear forms in the polynomial ring with $n+1$ variables. Indeed such an ideal can be related to the ideal of a collection of fat points, therefore it is possible to give a geometric interpretation of the conjecture in terms of linear systems with assigned multiple points. See Sect. 11.4.2 for details.

As one expects, when the multiplicities are big enough with respect to the degree, the conditions imposed by the multiple points are not linearly independent, as in the case of quadrics through double points. The above discussion in fact can be extended for instance to any linear system with degree d and points of multiplicity d: it is clear that any linear subspace $\mathbb{P}^r$ spanned by $r+1$ among the s base points is contained in the base locus of $\mathcal{L}$ with multiplicity d and gives a contribution to the speciality of $\mathcal{L}$ that depends on d and r.

In general it is challenging to compute the dimension of the linear systems. The issue is that the multiplicities of the points could force $\mathcal{L}$ to contain in the base locus, besides the multiple points, also higher dimensional cycles, the presence of which may generate speciality, as predicted in the Conjectures by Segre, Harbourne, Gimigliano and Hirschowitz for $\mathbb{P}^2$ [21, 22, 25, 32] (see also [12, 14, 15, 23]) and by Laface and Ugaglia for $\mathbb{P}^3$ [27].

In order to state the conjectures, we introduce the following notation. Given two planar linear systems $\mathcal{L} = \mathcal{L}_{2,d}(m_1, \dots, m_s)$ and $\mathcal{L}' = \mathcal{L}_{2,d'}(m'_1, \dots, m'_s)$ with the same s base points, we define their *intersection product* to be the intersection product of their strict transforms on the blow-up of $\mathbb{P}^2$ at the s points, namely the following number:

$$\mathcal{L}.\mathcal{L}' := dd' - \sum_{i=1}^{s} m_i m'_i.$$

If $\mathcal{L} = \mathcal{L}_{3,d}(m_1, \dots, m_s)$ is a linear system in $\mathbb{P}^3$ and L is the line spanned by the points p_i and p_j, then the *intersection product* of $\mathcal{L}$ and L is set to be the intersection product of their strict transforms in the blown-up of $\mathbb{P}^3$ at the points:

$$\mathcal{L}.L = d - m_i - m_j.$$

Conjecture 11.1 (Segre, Harbourne, Gimigliano, Hirschowitz) Fix $n = 2$. In the above notation, the linear system $\mathcal{L} = \mathcal{L}_{2,d}(m_1, \dots, m_s)$ is special if and only if there exists a rational curve in some linear system $\mathcal{L}' = \mathcal{L}_{2,d'}(m'_1, \dots, m'_s)$ such that $\mathcal{L}'.\mathcal{L}' = -1$ and $\mathcal{L}.\mathcal{L}' \leq -2$.

In spite of many partial results (see e.g. [12, 13] and references therein) −for instance it was proved to be true if $s \leq 9$ or $s = k^2$ with equal multiplicities [16, 20, 31] and for $m_i \leq 12$ and any number of points [14]− the conjecture is still open in general.

Assuming that Conjecture 11.1 holds for ten points in general position in $\mathbb{P}^2$, Laface and Ugaglia [27] formulated the following conjecture in the case $n = 3$, that has been proved to be true for $s \leq 8$ in [18].

Conjecture 11.2 (Laface, Ugaglia) Assume that $2d \geq m_{i_1} + m_{i_2} + m_{i_3} + m_{i_4}$, for any $\{i_1, i_2, i_3, i_4\} \subseteq \{1, \dots, s\}$. Then $\mathcal{L}$ is special if and only if one of the following holds:

1. There exists a line $L = \langle p_i, p_j \rangle$, for some $i, j \in \{1, \dots, s\}$ such that $\mathcal{L}.L \leq -2$;
2. There exists a quadric through 9 points $Q = \mathcal{L}_{3,2}(1^9)$ such that $Q.(\mathcal{L} - Q).(\mathcal{L} - K) < 0$, where K is the canonical divisor.

Remark 11.3 The arithmetic condition on the degree and the multiplicities in Conjecture 11.2 implies that $\mathcal{L}$ can not be reduced via some birational transformation of $\mathbb{P}^3$ – precisely via a cubo-cubic Cremona transformation – to a non-special linear system, see [27] for more details.

11.4 A Notion of Speciality for Linear Systems in $\mathbb{P}^n$

This section is devoted to the results obtained in [8].

In the direction of extending the existing conjectures for $n \leq 3$ to the case $n \geq 4$, and possibly to other projective varieties, a very natural and general question to address is the following.

Question 11.1 In the notation of Sect. 11.3, consider any non-empty linear system $\mathcal{L} = \mathcal{L}_{n,d}(m_1, \dots, m_s)$ in $\mathbb{P}^n$ and denote by D the corresponding divisor on the blow-up X of $\mathbb{P}^n$ at the s base points in general position. Let $\tilde{D}$ be the strict transform of D in the blow-up $\tilde{X}$ of X along the base locus of $\mathcal{L}$. Is $\tilde{D}$ non-special, namely, does $h^i(\tilde{X}, \mathcal{O}_{\tilde{X}}(\tilde{D}))$ vanish for all $i \geq 1$?

To answer this question one has to tackle two problems: the first one is to describe the base locus of $\mathcal{L}$, the second one is to understand the contribution given by each cycle (curve, surface, etc.) in the base locus to the speciality of $\mathcal{L}$, namely to $h^1(X, \mathcal{O}_X(D))$.

In the above notation, assume that $\mathcal{L}$ is a non-empty linear system. Let $I(r) \subseteq \{1, \dots, s\}$ be any multi-index consisting of $r + 1$ distinct indices, for $0 \leq r \leq \min(n, s) - 1$ and denote by $L_{I(r)} \cong \mathbb{P}^r$ the unique linear r-cycle through the points in general position p_i, for $i \in I(r)$. Define the number

$$k_{I(r)} := \max\left(\sum_{i \in I(r)} m_i - rd, 0\right).$$

The following result is easy to prove:

Lemma 11.1 (Linear Base Locus Lemma [8]) *Let $\mathcal{L} := \mathcal{L}_{n,d}(m_1, \dots, m_s)$ be a non-empty linear system. In the above notation, assume that $0 \leq r \leq n - 1$ and $k_{I(r)} > 0$. Then $\mathcal{L}$ contains in its base locus the cycle $L_{I(r)}$ with multiplicity at least $k_{I(r)}$.*

The proof is by induction on r. The case $r = 1$ of the statement is an easy consequence of Bézout's theorem.

The analysis of the linear part of the base locus yields a new definition of expected dimension [8, Definition 3.2]:

Definition 11.1 The *linear virtual dimension* of $\mathcal{L}$ is the number

$$\binom{n+d}{n} + \sum_{r=0}^{\min(n-1,s-1)} \sum_{I(r) \subseteq \{1,\dots,s\}} (-1)^{r+1} \binom{n + k_{I(r)} - r - 1}{n} - 1. \tag{11.1}$$

The *linear expected dimension* of $\mathcal{L}$, denoted by $\operatorname{ldim}(\mathcal{L})$, is the maximum of the linear virtual dimension of $\mathcal{L}$ and -1. A linear system $\mathcal{L}$ is said to be *linearly special* if $\dim(\mathcal{L}) \neq \operatorname{ldim}(\mathcal{L})$, *linearly non-special* otherwise.

If $\mathcal{L}$ is not empty, we have $\dim(\mathcal{L}) \geq \operatorname{ldim}(\mathcal{L}) \geq \operatorname{edim}(\mathcal{L})$, namely $\operatorname{ldim}(\mathcal{L})$ provides a better lower bound to the dimension of $\mathcal{L}$.

In (11.1), the number $(-1)^{r+1}\binom{n+k_{I(r)}-r-1}{n}$ computes the contribution of the linear cycle $L_{I(r)}$ spanned by the points p_j, $j \in I(r)$, which is contained in the base locus of $\mathcal{L}$ with multiplicity at least $k_{I(r)}$. For instance, in the case $\mathcal{L} = \mathcal{L}_{n,2}(2^s)$ discussed in Sect. 11.3, we have $k_{I(r)} = 2$, for all multi-indices $I(r) \subseteq \{1, \ldots, s\}$, $r \geq 1$. The contribution of each linear r-cycle $L_{I(r)}$ is $(-1)^{r+1}\binom{n-r+1}{n}$ and the speciality of $\mathcal{L}$ is completely described by the sum of these contributions: $\dim(\mathcal{L}) - \operatorname{edim}(\mathcal{L}) = \sum_{I(r),r\geq 1}(-1)^{r+1}\binom{n-r+1}{n}$. Therefore $\mathcal{L}$ is linearly non-special.

We address the following question:

Question 11.2 Classify the linearly special linear systems.

11.4.1 Classification Results

In what follows we give a complete answer to Question 11.2 and a partial answer to Question 11.1 for $s \leq n+2$. Moreover, for an arbitrary number of points $s \geq n+3$, we provide a sufficient condition for a linear system to be linearly non-special, partially answering Question 11.2. This summarizes the main results obtained in [8].

Case $s \leq n+2$

In [8, Section 4] a detailed description of the cohomologies of the strict transform $\tilde{D}$ of D with respect to the blow-up $\tilde{X}$ of $\mathbb{P}^n$ along the linear base locus is given. The results can be summarized as follows.

Denote by $\pi^n_{(0)} : X^n_{(0)} \to \mathbb{P}^n$ the blow-up of $\mathbb{P}^n$ at $p_1, \ldots, p_s$, with $E_1, \ldots, E_s$ exceptional divisors. Consider the following sequence of blow-ups:

$$X^n_{(n-1)} \xrightarrow{\pi^n_{(n-1)}} \cdots \xrightarrow{\pi^n_{(3)}} X^n_{(2)} \xrightarrow{\pi^n_{(2)}} X^n_{(1)} \xrightarrow{\pi^n_{(1)}} X^n_{(0)},$$

where $X^n_{(r)} \xrightarrow{\pi^n_{(r)}} X^n_{(r-1)}$ denotes the blow-up of $X^n_{(r-1)}$ along the union of the pull-backs of the linear subspaces $L_{I(r)} \subset \mathbb{P}^n$, via $\pi^n_{(r-1)} \circ \cdots \circ \pi^n_{(0)}$. Let $E_{I(r)}$ be the corresponding exceptional divisors. We will denote, abusing notation, by H the pull-back in $X^n_{(r)}$ of the hyperplane class of $\mathbb{P}^n$ and by $E_{I(\rho)}$, for $0 \leq \rho \leq r-1$, the pull-backs in $X^n_{(r)}$ of the exceptional divisors of $X^n_{(\rho)}$, respectively. The Picard group of $X^n_{(r)}$ is therefore $\operatorname{Pic}(X^n_{(r)}) = \langle H, E_{I(\rho)} : p \leq \rho \leq r\rangle$.

The strict transform of $D = D_{(0)} \in \mathrm{Pic}(X^n_{(0)})$ via the composed map $\pi^n_{(r)} \circ \cdots \circ \pi^n_{(0)}$ is the following divisor

$$D_{(r)} = dH - \sum_{I(\rho),\ 0\le\rho\le r} k_{I(\rho)} E_{I(\rho)} \in \mathrm{Pic}(X^n_{(r)})$$

Abusing notation, we will abbreviate with $D_{(r)}$ the sheaf $\mathcal{O}_{X^n_{(r)}}(D_{(r)})$.

Theorem 11.1 (Brambilla, Dumitrescu, Postinghel [8]) *In the notation above, for every r such that $1 \le r \le n-1$, then*

$$h^{r+1}(D_{(r)}) = \sum_{I(\rho),\ r+1\le\rho\le\min\{s-1,n-1\}} (-1)^{r+1-\rho} \binom{n + k_{I(\rho)} - \rho - 1}{n}$$

and $h^i(D_{(r)}) = 0, i \ne 0, r+1$. In particular $\dim(\mathcal{L}) = \mathrm{ldim}(\mathcal{L})$ *and the speciality of $\mathcal{L}$ is given by*

$$h^1(\mathcal{L}) = \sum_{I(r),\ 1\le r\le\min\{s-1,n-1\}} (-1)^{r-1} \binom{n + k_{I(r)} - r - 1}{n}.$$

In the first part of Theorem 11.1 it is proved that an r-dimensional linear cycle $L_{I(r)}$ for which $k_{I(r)} \ge 1$ gives a contribution, that is $(-1)^{r+1}\binom{n+k_{I(r)}-r-1}{n}$, at the level of the r-th cohomology group of the strict transform $D_{(r-1)}$ of D, after blowing-up all cycles of dimension at most $r-1$. The second part furnishes a partial answer to Question 11.1.

A consequence of this result is that any non-empty linear system $\mathcal{L}_{n,d}(m_1,\ldots,m_s)$ in $\mathbb{P}^n$ is always linearly non-special if $s \le n+2$. In particular this gives a positive answer to Question 11.1 and it shows that the Laface-Ugaglia Conjecture for linear systems of $\mathbb{P}^3$ holds in this range.

The proof is by induction on r and n and is based on the computation of the Euler characteristic, denoted simply by $\chi(D_{(r)})$, of the sheaves $\mathcal{O}_{X^n_{(r)}}(D_{(r)})$, for $0 \le r \le \min\{s-1, n-1\}$. We discussed above the equality $\chi(D_{(0)}) - 1 = \mathrm{vdim}(\mathcal{L})$, which is the expected value for the dimension of $\mathcal{L}$, namely the number of polynomials of degree d minus the number of linear conditions imposed by the multiple points, see Sect. 11.3. For $r = 1$ the Euler characteristic also detects the linear obstructions to the non-speciality of $\mathcal{L}$ given by the presence of multiple lines in the base locus of $\mathcal{L}$, that is $\chi(D_{(1)}) = \chi(D_{(0)}) + \sum_{I(1)}\binom{n+k_{I(1)}-2}{n}$. For $r = 2$ also the presence of planes is detected, namely $\chi(D_{(2)}) = \chi(D_{(1)}) - \sum_{I(2)}\binom{n+k_{I(2)}-3}{n}$, and so on. The Euler characteristic of $D_{(\bar{r})}$ encodes the (alternating) sum of all contributions to the speciality of $\mathcal{L}$ given by linear cycles, in particular $\chi(D_{(\bar{r})}) - 1$ equals the number (11.1) introduced in Definition 11.1. The interested reader can find details in [8, Section 4.3].

Case $s \geq n+3$

For the case $s \geq n+3$, in [8] a sufficient condition for a linear system with an arbitrary number of general points to be linearly non-special is given. More precisely, we prove that if the sum of the multiplicities of an arbitrary number $s \geq n+3$ of points is bounded with respect to the degree d, then the speciality of $\mathcal{L}$ is completely described by the linear obstructions, namely by the linear cycles $L_{I(r)} \cong \mathbb{P}^r$ spanned by the base points which are contained at least doubly in the base locus and the dimension of $\mathcal{L}$ is therefore computed.

Let $s(d) \geq 0$ be the number of points of multiplicity d, set $b(\mathcal{L}) := \min\{n - s(d), s-n-2\}$ and consider the inequality

$$\sum_{i=1}^{s} m_i \leq nd + b(\mathcal{L}). \tag{11.2}$$

Theorem 11.2 (Brambilla, Dumitrescu, Postinghel [8]) *In the notation above, assume that $\mathcal{L}$ satisfies condition (11.2). Then* $\dim(\mathcal{L}) = \operatorname{ldim}(\mathcal{L})$.

In particular this proves that the Laface-Ugaglia Conjecture holds if $n=3$ and the linear system satisfies (11.2).

The proof is based on an adaptation of the degeneration technique introduced by Hirschowitz [24], *la méthode d'Horace*, which consists in making iterated specializations of as many points as convenient on a fixed hyperplane H and then applying induction on n and d. One gets the so called Castelnuovo exact sequence:

$$0 \longrightarrow \hat{\mathcal{L}} \longrightarrow \mathcal{L} \longrightarrow \mathcal{L}_{|H} \longrightarrow 0,$$

where the kernel $\hat{\mathcal{L}}$ is a linear system of hypersurfaces of degree $d-1$ and s points with lower multiplicities, while the restricted system $\mathcal{L}_{|H}$ is a linear system of degree-d hypersurfaces with multiple points in $\mathbb{P}^{n-1}$. Thus, arguing by induction, if the two external systems are linearly non-special and non-empty, then the system $\mathcal{L}$ is linearly non-special too, because $\dim(\mathcal{L}) = \dim(\hat{\mathcal{L}}) + \dim(\mathcal{L}_{|H}) + 1$ and $\operatorname{ldim}(\mathcal{L}) = \operatorname{ldim}(\hat{\mathcal{L}}) + \operatorname{ldim}(\mathcal{L}_{|H}) + 1$.

11.4.2 Final Remarks

The linear expected dimension $\operatorname{ldim}(\mathcal{L})$ is meant to be a refined version of the expected dimension $\operatorname{edim}(\mathcal{L})$. Indeed, in the notation introduced above, $\operatorname{edim}(\mathcal{L}) = \max(\chi(D_{(0)}), 0) - 1$ and $\operatorname{ldim}(\mathcal{L}) = \max(\chi(D_{(\bar{r})}), 0) - 1$, and the difference of these numbers encodes the contributions of the linear obstructions to the speciality of $\mathcal{L}$.

Nevertheless, not only linear cycles contained with multiplicity in the base locus of a linear system generate speciality. For instance the existence in the base locus of multiple rational normal curves, plays an important role. We remark that when the points are $s \geq n+3$, the assumption (11.2) is, in particular, a sufficient condition for the base locus to contain no multiple rational normal curves through $n+3$ points. In fact, we expect that when multiple rational normal curves appear in the base locus, they give a contribution to the speciality of the system.

On the other hand, as already noticed by Laface and Ugaglia for the case $n = 3$, also the existence of quadric surfaces passing through nine general points in the base locus can give contribution to the speciality of a linear system. In this case it does not seem very clear how to quantify such contributions.

It would be interesting to extend the definition of linear expected dimension of a linear system on $\mathbb{P}^n$ taking into account also the contribution of the non-linear positive dimensional cycles contained with multiplicity in the base locus. This may lead to a natural generalization to the case $n \geq 4$ of Conjecture 11.1 for $\mathbb{P}^2$ and Conjecture 11.2 for $\mathbb{P}^3$.

Connection to the Fröberg-Iarrobino Conjecture

The dimensionality problem for linear systems with assigned multiple points is related to the Fröberg-Iarrobino Conjecture, which gives a predicted value for the Hilbert series of an ideal generated by s general d-powers of linear forms in the polynomial ring with $n+1$ variables. In terms of our Definition 11.1 the conjecture can be stated as follows: a linear system is always linearly non-special but in a finite list of exceptions, see [8, Sect. 6.1] for more details.

In [11, Proposition 9.1] Chandler proves that the Fröberg-Iarrobino Conjecture is true if either $s \leq n+1$ or $\sum_{i=1}^{s} m_i \leq dn+1$. Theorems 11.1 and 11.2 improve Chandler's result and show that the Fröberg-Iarrobino conjecture holds if either $s \leq n+2$ or condition (11.2) is satisfied.

11.5 Secant Varieties and Linear Systems with Double Points

Let X be a non-degenerate complex projective variety of dimension n embedded in $\mathbb{P}^N$. The *s-secant variety* $\mathrm{Sec}_s(X)$ of X is defined to be the Zariski closure of the union of the linear spans in $\mathbb{P}^N$ of s-tuples of independent points of X. By a parameter computation one gets that

$$\dim(\mathrm{Sec}_s(X)) \leq \mathrm{edim}(\mathrm{Sec}_s(X)) := \min\{sn+s-1, N\},$$

where the integer on the right hand side is called the *expected dimension* of $\mathrm{Sec}_s(X)$. The variety X is said to be *s-defective* if $\dim(\mathrm{Sec}_s(X)) < \mathrm{edim}(\mathrm{Sec}_s(X))$.

The d-th *Veronese embedding* $X_{n,d}$ of the projective space $\mathbb{P}^n$ is the map $\nu_{n,d} : \mathbb{P}^n \to \mathbb{P}^{\binom{n+d}{n}-1}$ given by

$$[x_0 : x_1 : \cdots : x_n] \mapsto [x_0^d : x_0^{d-1}x_1 : \cdots : x_n^d].$$

The *Segre embedding* of the product $\mathbb{P}^{n_1} \times \mathbb{P}^{n_2}$ is the map Seg : $\mathbb{P}^{n_1} \times \mathbb{P}^{n_2} \to \mathbb{P}^{(n_1+1)(n_2+1)-1}$ defined as

$$([x_0 : x_1 : \cdots : x_{n_1}], [y_0 : y_1 : \cdots : y_{n_2}]) \mapsto [x_0y_0 : x_0y_1 : \cdots : x_{n_1}y_{n_2}].$$

Combining the two maps one obtains the so called *Segre-Veronese embeddings* of products of projective spaces as follows. Fix integers $r \geq 1$, $n_1, \ldots, n_r \geq 1$ and $d_1, \ldots, d_r \geq 1$. Set $N := \prod_{i=1}^{r} \binom{n_i+d_i}{n_i} - 1$, $\mathbf{n} = (n_1, \ldots, n_r)$ and $\mathbf{d} = (d_1, \ldots, d_r)$. The multi-degree-$\mathbf{d}$ Segre-Veronese embedding of $\mathbb{P}^{\mathbf{n}} := \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$ is the map $\nu_{\mathbf{n},\mathbf{d}} : \mathbb{P}^{\mathbf{n}} \to \mathbb{P}^N$ defined as follows:

$$([x_{1,0} : \cdots : x_{1,n_1}], \ldots, [x_{r,0} : \cdots : x_{r,n_r}]) \mapsto [\cdots : \prod_{i=1}^{r} \prod_{j=0}^{n_i} x_{i,j}^{d_{i,j}} : \cdots],$$

with $\sum_{j=0}^{n_i} d_{i,j} = d_i$, $i = 1, \ldots, r$. Denote by $X_{\mathbf{n},\mathbf{d}} \subseteq \mathbb{P}^N$ the Zariski closure of the image of $\nu_{\mathbf{n},\mathbf{d}}$ that we will call *Segre-Veronese variety*. Notice that the map $\nu_{\mathbf{n},\mathbf{1}}$ is the Segre embedding of $\mathbb{P}^{\mathbf{n}}$.

Secant varieties of Segre-Veronese varieties are not well-understood so far. The problem of determining the dimension of the s-secant varieties of the Segre-Veronese varieties $X_{\mathbf{n},\mathbf{d}}$ is very hard and is open in general. Several partial results are known for Segre-Veronese varieties of small dimension [1, 5, 7, 10], and in any dimension for Veronese varieties by the Alexander-Hirschowitz Theorem [3, 9, 29] (cf. Sect. 11.3). A conjectural classification of secant defective Segre varieties is given in [2, Question 6.6]. This conjecture was proved to be true for $\mathbf{n} = (1, \ldots, 1)$ in [6] and for $s \leq 6$ in [2], but it is still open in general.

Tensor Decomposition and the Waring Problems

Secant varieties of Segre-Veronese varieties are of particular interest because of their connection to tensor decomposition problems. Tensors play a wider and wider role in numerous applications including signal processing [17], the study of entanglement in quantum physics [19], phylogenetics [4] and many others. For secant varieties there is a variant of the notion of rank of a tensor, namely the border rank, that turns out to be extremely useful in applications as one is often more interested in the limiting rank of a tensor rather than in its exact rank.

For a non-degenerate projective variety $X \subset \mathbb{P}^N$, the X-*rank* of a point $p \in \mathbb{P}^N$, $R_X(p)$, is the smallest integer s such that p belongs to the linear span of s points of X. The X-*border rank* of $p \in \mathbb{P}^N$, $\underline{R}_X(p)$, is the smallest s such that p belongs to the limit of linear spans of s points of X, namely if $p \in \mathrm{Sec}_s(X)$. If X is a

Segre-Veronese variety, the X-rank and X-border rank agree with the notions of rank and border rank of partially symmetric tensors. See [28].

The classical problem of determining the X-rank of a point in a specific situation is often referred to as the *generalized Waring problem* in tribute to Waring who asked in the eighteenth century about the presentation of an integer as a sum of powers: Given positive integers d and s, may we write any positive integer as a sum of s non-negative d-th powers? The *Waring problem for polynomials* is: Given positive integers d, s, n, what is the minimal $s = s(d, n)$ such that a general homogeneous polynomial $f(x_0, \dots, x_n)$ of degree d can be expressed as a sum of s d-th powers of linear forms $l_i(x_0, \dots, x_n)$, $i = 1, \dots, s$? The image $X_{n,d}$ of the d-Veronese embedding of $\mathbb{P}^n$ is the set of (projectivized) d-th powers of linear forms and $\mathrm{Sec}_s(X_{n,d})$ is the Zariski closure of the set of homogeneous polynomials that can be written as the sum of s d-th powers of linear forms. Therefore the Waring problem for polynomials can be restated in the following way: What is the smallest s such that $\mathrm{Sec}_s(X_{n,d}) = \mathbb{P}^N$? The solution to the problem is given in the Alexander-Hirschowitz Theorem, that we previously discussed. Indeed one can rephrase the problem of determining the dimension of the s-secant varieties of the Veronese embeddings in terms of linear systems of $\mathbb{P}^n$ with imposed double points. We explain this correspondence in the more general case of Segre-Veronese embeddings in the next section.

Rephrasing in Terms of Prescribing Double Points

Computing the dimension of the s-secant variety of a Segre-Veronese variety is equivalent to calculating the dimension of the linear systems $\mathcal{L}_{\mathbf{n},\mathbf{d}}(2^s)$ of multi-degree $\mathbf{d}$ hypersurfaces of $\mathbb{P}^{\mathbf{n}}$ that are singular at s points in general position. This is a consequence of a classical result, known as *Terracini's Lemma*.

Lemma 11.2 (Terracini's Lemma) *Let $X \subseteq \mathbb{P}^N$ be an irreducible, non-degenerate, projective variety. Let $p_1, \dots, p_s$ be general points of X, with $s \leq N + 1$. Then the tangent space to $Sec_s(X)$ at a general point $q \in \langle p_1, \dots, p_s \rangle$ equals the linear span of the tangent spaces to X at $p_1, \dots, p_s$:*

$$T_{Sec_s(X),q} = \left\langle T_{X,p_1}, \dots, T_{X,p_s} \right\rangle.$$

A multi-degree $\mathbf{d}$ hypersurface S of $\mathbb{P}^{\mathbf{n}}$ corresponds, via the Segre-Veronese embedding $\nu_{\mathbf{n},\mathbf{d}}$, to a hyperplane section H of $X_{\mathbf{n},\mathbf{d}} \subseteq \mathbb{P}^N$. Moreover S has a double point at p if and only if H is tangent to $X_{\mathbf{n},\mathbf{d}}$ at $\nu_{\mathbf{n},\mathbf{d}}(p)$. Now, fix $p_1, \dots, p_s$ general points in $\mathbb{P}^{\mathbf{n}}$ and consider the linear system $\mathcal{L}_{\mathbf{n},\mathbf{d}}(2^s)$ of multi-degree $\mathbf{d}$ hypersurfaces which are singular at $p_1, \dots, p_s$. It is in correspondence with the linear system of hyperplane sections in $\mathbb{P}^N$ tangent to $X_{\mathbf{n},\mathbf{d}}$ at $\nu_{\mathbf{n},\mathbf{d}}(p_1), \dots, \nu_{\mathbf{n},\mathbf{d}}(p_s)$, namely with the intersection $\bigcap_{i=1}^{s}(T_{X_{\mathbf{n},\mathbf{d}},\nu_{\mathbf{n},\mathbf{d}}(p_i)})^{\perp}$ of the orthogonal spaces to the tangent spaces. Furthermore, since

$$(T_{X_{\mathbf{n},\mathbf{d}},\nu_{\mathbf{n},\mathbf{d}}(p_1)})^{\perp} \cap \cdots \cap (T_{X_{\mathbf{n},\mathbf{d}},\nu_{\mathbf{n},\mathbf{d}}(p_s)})^{\perp} = \left\langle T_{X_{\mathbf{n},\mathbf{d}},\nu_{\mathbf{n},\mathbf{d}}(p_1)}, \dots, T_{X_{\mathbf{n},\mathbf{d}},\nu_{\mathbf{n},\mathbf{d}}(p_s)} \right\rangle^{\perp},$$

then by Lemma 11.2 one gets

$$\dim(\mathrm{Sec}_s(X_{\mathbf{n},\mathbf{d}})) = \dim(T_{\mathrm{Sec}_s(X_{\mathbf{n},\mathbf{d}}),q}) = N - \dim(\mathcal{L}_{\mathbf{n},\mathbf{d}}(2^s)),$$

where q is a general point of the secant variety $\mathrm{Sec}_s(X_{\mathbf{n},\mathbf{d}})$.

11.6 Classification of Special Linear Systems with Double Points in $(\mathbb{P}^1)^n$

This section is devoted to the results obtained in [26].

A natural approach to the dimensionality problem of linear systems is via degenerations. Degenerations allow one to move the multiple base points of a linear system in special position, using a semi-continuity argument. Ciliberto and Miranda in [14] and [15] exploited a degeneration of the plane, originally proposed by Ran [30] to study higher multiplicity interpolation problems for planar linear systems with general multiple base points. This approach consists in degenerating the plane to a reducible surface, with two components intersecting along a line, and simultaneously degenerating the linear system to a limit linear system which is somewhat easier than the original one. In particular this degeneration argument allows to use induction either on the degree or on the number of imposed multiple points. This method was generalized in [29] to the case of $\mathbb{P}^n$ to study linear systems of degree d hypersurfaces with a general collection of double points and provide a short and simplified proof of the Alexander-Hirschowitz Theorem. A further generalization is used in [26] to classify linear systems of multi-degree $\mathbf{d}$ hypersurfaces of $\mathbb{P}^1 \times \cdots \times \mathbb{P}^1$ with assigned general double points.

11.6.1 Classification Results

Let $\mathcal{L}_{\mathbf{d}}(2^s)$ be the linear system of multi-degree $\mathbf{d}$ hypersurfaces of $(\mathbb{P}^1)^n = \mathbb{P}^1 \times \cdots \times \mathbb{P}^1$ with s prescribed general nodes. In [26] the following complete classification result is proved.

Theorem 11.3 (Laface, Postinghel [26]) *The linear system $\mathcal{L}_{\mathbf{d}}(2^s)$ of $(\mathbb{P}^1)^n$ is non-special except in the following cases.*

n	degrees	s	$\mathrm{edim}(\mathcal{L})$	$\dim(\mathcal{L})$
2	$(2, 2a)$	$2a + 1$	-1	0
3	$(1, 1, 2a)$	$2a + 1$	-1	0
3	$(2, 2, 2)$	7	-1	0
4	$(1, 1, 1, 1)$	3	0	1

The proof, by induction on the number of factors n and on the multi-degree $\mathbf{d}$, exploits a degeneration of $(\mathbb{P}^1)^n$ and of $\mathcal{L}_{\mathbf{d}}(2^s)$ as described in Sect. 11.6.2. A basic step for the induction is represented in the fundamental paper [6], where the authors show that, if all the $d_i = 1$, then $\mathcal{L}_{(1,\dots,1)}(2^s)$ has always but in one case ($n = 4$) the expected dimension.

This theorem's reformulation in terms of secant varieties provides a complete classification of all defective secant varieties of Segre-Veronese embeddings $X_{\mathbf{d}} \subseteq \mathbb{P}^N$, $N = \prod_{i=1}^{n}(d_i + 1) - 1$, of products of projective lines.

Theorem 11.4 (Laface, Postinghel [26]) *The s-secant variety of $X_{\mathbf{d}} \subseteq \mathbb{P}^N$ is non-defective with the list of exceptions of Theorem 11.3.*

11.6.2 Degeneration Techniques

The toric degeneration of the Segre-Veronese embedding $X_{\mathbf{d}}$ of $(\mathbb{P}^1)^n$ employed in [26] is the following.

Degeneration of the Variety

Let $P = P_{\mathbf{d}}$ be the convex lattice polytope $[0, d_1] \times \cdots \times [0, d_n] \subseteq \mathbb{R}^n$. Its integer points define the toric map which is the Segre-Veronese embedding $\nu_{\mathbf{d}} : (\mathbb{P}^1)^n \to X_{\mathbf{d}} \subseteq \mathbb{P}^N$. Fix an integer k such that $1 \leq k \leq d_n - 1$ and consider the function $\phi : P \cap \mathbb{Z}^n \to \mathbb{Z}$ defined by

$$\phi(v) = \begin{cases} 0 & \text{if } v_n \leq k, \\ v_n - k & \text{if } v_n > k. \end{cases}$$

It defines a *regular subdivision* of P in the following way. The convex hull of the half lines $\{(v, t) \in P \times \mathbb{R}_{\geq 0} : t \geq \phi(v)\}$ is an unbounded polyhedron with two lower faces. By projecting these faces onto P one obtains the regular subdivision $\{P^1, P^2\}$ of P, where $P^1 = P_{(d_1,\dots,d_{n-1},d_n-k)}$, $P^2 = P_{(d_1,\dots,d_{n-1},k)}$ and $P^1 \cap P^2 = P_{(d_1,\dots,d_{n-1})} = [0, d_1] \times \cdots \times [0, d_{n-1}]$. We show the configuration of this toric degeneration in Fig. 11.1 for $n = 3$.

The regular subdivision defines a 1-dimensional embedded degeneration, i.e. a 1-parameter family $\{X_t\}_{t\in\mathbb{C}}$ of varieties, whose general fiber X_t, $t \neq 0$, is isomorphic to $X_{\mathbf{d}}$ and whose central fiber X_0 is isomorphic to the union $X^1 \cup X^2$, where X^1 is the Segre-Veronese embedding $X_{(d_1,\dots,d_{n-1},d_n-k)} \subseteq \mathbb{P}^{N^1}$ of $(\mathbb{P}^1)^n$ and, similarly, X^2 is the Segre-Veronese embedding $X_{(d_1,\dots,d_{n-1},k)} \subseteq \mathbb{P}^{N^2}$, where $N^1 := \prod_{i=1}^{n-1}(d_i + 1)(d_n - k + 1) - 1$ and $N^2 := \prod_{i=1}^{n-1}(d_i + 1)(k + 1) - 1$. The intersection $X^{1,2} := X^1 \cap X^2$ is the Segre-Veronese embedding $X_{(d_1,\dots,d_{n-1})} \subseteq \mathbb{P}^{N^{1,2}}$ of $(\mathbb{P}^1)^{n-1}$, where $N^{1,2} := \prod_{i=1}^{n-1}(d_i + 1) - 1$.

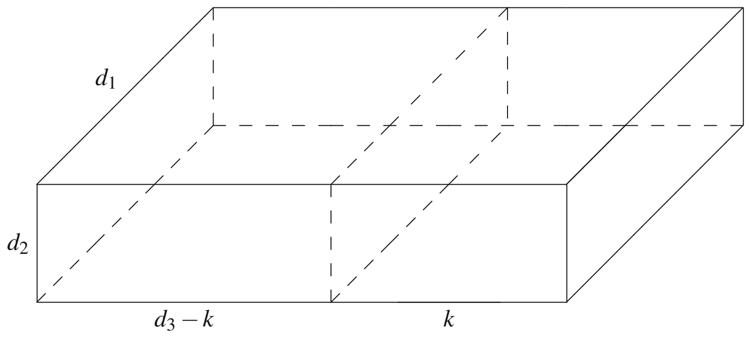


Fig. 11.1 A regular subdivision of $P_{(d_1,d_2,d_3)}$

Degenerating the Linear System

We consider the linear system $\mathcal{L}_t := \mathcal{L} = \mathcal{L}_{\mathbf{d}}(2^s)$ with s assigned general double points $p_{1,t}, \dots, p_{s,t}$, $t \neq 0$. A linear system on X_0 is given by two linear systems, respectively on X^1 and on X^2, which agree on the intersection $X^{1,2}$.

Fix a non-negative integer $s^1 \leq s$ and specialize s^1 points generically on X^1 and the other $s^2 := s - s^1$ points generically on X^2, i.e. take a flat family $\{p_{1,t} \dots, p_{s,t}\}_{t\in\mathbb{C}}$ such that $p_{1,0}, \dots, p_{s^1,0} \in X^1$ and $p_{s^1+1,0}, \dots, p_{s,0} \in X^2$. The limiting linear system $\mathcal{L}_0$ on X_0 is formed by the flat limits of the elements of multi-degree $\mathbf{d}$ on the general fiber X_t which are singular at $p_{1,0}, \dots, p_{s,0}$. Consider the following linear systems:

$$\begin{aligned} \mathcal{L}^1 &:= \mathcal{L}_{(d_1,\dots,d_{n-1},d_n-k)}(2^{s^1}), & \mathcal{L}^2 &:= \mathcal{L}_{(d_1,\dots,d_{n-1},k)}(2^{s^2}), \\ \hat{\mathcal{L}}^1 &:= \mathcal{L}_{(d_1,\dots,d_{n-1},d_n-k-1)}(2^{s^1}), & \hat{\mathcal{L}}^2 &:= \mathcal{L}_{(d_1,\dots,d_{n-1},k-1)}(2^{s^2}), \end{aligned} \tag{11.3}$$

where $\mathcal{L}^i$, $\hat{\mathcal{L}}^i$ are defined on X^i and $\hat{\mathcal{L}}^i$ is the kernel of the restriction map of $\mathcal{L}^i$ to $X^{1,2}$, $i = 1, 2$. This is given by the exact sequence:

$$0 \longrightarrow \hat{\mathcal{L}}^i \longrightarrow \mathcal{L}^i \longrightarrow \mathcal{L}^i{}_{|X^{1,2}} \longrightarrow 0.$$

We have

$$\mathcal{L}_0 := \mathcal{L}^1 \times_{\mathcal{L}^{1,2}} \mathcal{L}^2, \quad \mathcal{L}^{1,2} := \mathcal{L}^1{}_{|X^{1,2}} \cap \mathcal{L}^2{}_{|X^{1,2}}$$

and, by upper semi-continuity, that $\dim(\mathcal{L}_0) \geq \dim(\mathcal{L}_t)$.

Lemma 11.3 *In the above notation, if* $\dim(\mathcal{L}_0) = \operatorname{edim}(\mathcal{L})$*, then the linear system* $\mathcal{L}$ *has the expected dimension, i.e. it is non-special.*

By choosing suitable integers s^1, s^2 and k we prove Theorem 11.3.

Acknowledgements I wish to express my gratitude to the coordinator Dr. Tor Dokken and all members of the SAGA network for creating an excellent working atmosphere, and to my mentor Prof. Ragni Piene for her valuable guidance. I would like to thank the Centre of Mathematics for Applications (CMA, University of Oslo) for providing me with quality work environment and facilities.

References

1. H. Abo, M.C. Brambilla, On the dimensions of secant varieties of Segre-Veronese varieties. Ann. Mat. Pura Appl. (4) **192**(1), 61–92 (2013)
2. H. Abo, G. Ottaviani, C. Peterson, Induction for secant varieties of Segre varieties. Trans. Am. Math. Soc. **361**(2), 767–792 (2009)
3. J. Alexander, A. Hirschowitz, Polynomial interpolation in several variables. J. Algebr. Geom. **4**(2), 201–222 (1995)
4. Allman, E. S., J.A. Rhodes, Phylogenetic ideals and varieties for the general Markov model. Adv. Appl. Math. **40**(2), 127–148 (2008)
5. E. Ballico, A. Bernardi, M.V. Catalisano, Higher secant varieties of $\mathbb{P}^n \times \mathbb{P}^1$ embedded in bi-degree (a, b). Commun. Algebra **40**(10), 3822–3840 (2012)
6. M. V. Catalisano, A. V. Geramita, A. Gimigliano, Secant varieties of $\mathbb{P}61 \times \cdots \times \mathbb{P}^1$ (n-times) are not defective for $n \geq 5$. J. Algebraic Geometry **20**(2), 295–327 (2011)
7. A. Bernardi, E. Carlini, Gimigliano, M. V.: Higher secant varieties of $\mathbb{P}^n \times \mathbb{P}^m$ embedded in bi-degree $(1, d)$. J. Pure Appl. Algebra **215**(12), 2853–2858 (2011)
8. M.C. Brambilla, O. Dumitrescu, E. Postinghel, On a notion of speciality of linear systems in $\mathbb{P}^n$. Trans. Am. Math. Soc. (2014, to appear)
9. M.C. Brambilla, O. Ottaviani, On the Alexander-Hirschowitz theorem. J. Pure Appl. Algebra **212**(5), 1229–1251 (2008)
10. J. Buczyński, J.M. Landsberg, Ranks of tensors and a generalization of secant varieties. Linear Algebra Appl. **438**(2), 668–689 (2013)
11. K.A. Chandler, The geometric interpretation of Fröberg-Iarrobino conjectures on infinitesimal neighbourhoods of points in projective space. J. Algebra **286**(2), 421–455 (2005)
12. C. Ciliberto, Geometric aspects of polynomial interpolation in more variables and of Waring's problem, in *Proceedings of the Third European Congress of Mathematics, I, (Barcelona 2000).* Progress in Mathemetics, vol. 201 (Birkhäuser, Basel, 2001), pp. 289–316
13. C. Ciliberto, B. Harbourne, R. Miranda, J. Roé, *Variations on Nagata's Conjecture*. Clay Mathematics Proceedings, vol. 18 (American Mathematical Society, Providence, 2013), pp. 185–203
14. C. Ciliberto, R. Miranda, Degenerations of planar linear systems. J. Reine Angew. Math. **501**, 191–220 (1998)
15. C. Ciliberto, R. Miranda, Linear systems of plane curves with base points of equal multiplicity. Trans. Am. Math. Soc. **352**(9), 4037–4050 (2000)
16. C. Ciliberto, R. Miranda, Nagata's conjecture for a square or nearly-square number of points. Ric. Mat. **55**(1), 71–78, (2006)
17. L. De Lathauwer, J. Castaing, Tensor-based techniques for the blind separation of DS-CDMA signals. Signal Process. **87**(2), 322–336 (2007)
18. C. De Volder, A. Laface, On linear systems of $\mathbb{P}^3$ through multiple points. J. Algebra **310**(1), 207–217 (2007)
19. J. Eisert, D. Gross, Multiparticle entanglement, in *Physics Textbook*, ed. by D. Bruß et al. Lectures on Quantum Information (Wiley-VCH, Weinheim, 2007), pp. 237–252
20. L. Evain, Une minoration du degré des courbes planes à singularités imposées. Bull. S.M.F. **126**, 525–543 (1998)
21. A. Gimigliano, On linear systems of plane curves, Ph.D. thesis, Queen's University, 1987

22. B. Harbourne, The geometry of rational surfaces and Hilbert functions of points in the plane. Can. Math. Soc. Conf. Proc. **6**, 95–111 (1986)
23. B. Harbourne, Points in good position in $\mathbf{P}^2$, in *Zero-dimensional Schemes*, Ravello, 1992 (de Gruyter, Berlin, 1994), pp. 213–229
24. A. Hirschowitz, La méthode d'Horace pour l'interpolation á plusiers variables. Manuscr. Math. **50**, 1091–1110 (1985)
25. A. Hirschowitz, Une conjecture pour la cohomologie des diviseurs sur les surfaces rationnelles génériques. J. Reine Angew. Math. **397**, 208–213 (1989)
26. A. Laface, E. Postinghel, Secant varieties of Segre-Veronese embeddings of $(\mathbb{P}^1)^r$. Math. Ann. **356**(4), 1455–1470 (2013)
27. A. Laface, L. Ugaglia, On a class of special linear systems of $\mathbb{P}^3$. Trans. Am. Math. Soc. **358**(12), 5485–5500 (2006)
28. J.M. Landsberg, *Tensors: Geometry and Applications*. Graduate Studies in Mathematics, vol. 128 (American Mathematical Society, Providence, 2012)
29. E. Postinghel, A new proof of the Alexander-Hirschowitz interpolation theorem. Ann. Mat. Pura Appl. (4) **191**(1), 77–94 (2012)
30. Z. Ran, Enumerative geometry of singular plane curves. Invent. Math. **9**, 447–465 (1989)
31. J. Roè, Limit linear systems and applications (2006, preprint). http://arxiv.org/abs/math.AG/0602213
32. B. Segre, Alcune questioni su insiemi finiti di punti in geometria algebrica. Atti Convegno Intern. di Geom. Alg. di Torino 15–33 (1961)

Chapter 12
Rational Parametrizations of Edge and Corner Blends for Isogeometric Analysis

Heidi E.I. Dahl

12.1 Introduction

One of the major bottlenecks in the traditional computer aided design-analysis-redesign cycle is the transition between design and analysis models: the tools used in Computer Aided Design (CAD) and those used in Finite Element Analysis (FEA) have been developed independently, and their model representations have been chosen based on different needs and priorities. Isogeometric Analysis (IGA) seeks to address this by using the same geometric model throughout, from which both analysis models and design models can be extracted (see, e.g., [4]).

When developing geometric models for IGA we need to reconcile the different requirements of CAD and FEA. For example, though shape accuracy is important in CAD, gaps between adjacent elements are allowed within fine tolerances. In FEA, however, adjacent elements are required to match exactly. The introduction of IGA has therefore led to a renewed interest in exact representations of curve, surface and volume elements.

In CAD, and in particular in the design of mechanical parts, complex shapes are to a large extent constructed from planes, the natural quadrics (spheres, and right circular cylinders and cones), and rolling ball blends between them. When the rolling ball blends are rational, these are all Pythagorean Normal (PN) surfaces: rational surfaces with rational unit normal vector fields. By extending the surface parametrizations along the unit normal vectors, we can construct thick surfaces, a relatively simple class of rational volume parametrizations. Furthermore, if the unit normal vector fields of two adjacent surface patches match along the common boundary curve, the two volume elements will match exactly along the resulting

H.E.I. Dahl (✉)
Department of Applied Mathematics, SINTEF ICT, Forskningsveien 1, 0373 Oslo, Norway
e-mail: heidi.dahl@sintef.no

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_12

boundary surface. These volume parametrizations are therefore well-suited for FEA, and thus for IGA.

In previous papers [5, 6] we have described algorithms for minimal bi-degree rational parametrizations of fixed and variable radius rolling ball blends of the natural quadrics. In this paper we will outline several strategies for constructing variable radius blends between two primitive surfaces, and show how such blends can be combined to blend a given configuration of edges and corners. We will also specify the level of continuity between the individual blending surface patches. For simplicity we will focus on corners with three faces. However, some of our results can be generalized to n-sided corners.

The chapter is structured as follows: Sect. 12.2 outlines the motivation for investigating variable radius rolling ball blends, and Sect. 12.3 introduces the key concepts from Laguerre geometry used in our constructions. Section 12.4 presents our methods for constructing variable radius edge and corner blends, and Sect. 12.5 shows how they are applied to blend a composite corner example. Section 12.6 outlines how our approach may be generalized to PN surfaces, before we summarize our results in the "Conclusions" section

12.2 Beyond Fixed Radius Blends

The primitive surfaces most commonly used in CAD, i.e., planes and the natural quadrics, are PN surfaces. This gives us several desirous properties when creating surface and volume parametrizations. But although planes and natural quadrics are rational surfaces, this is not always the case for rolling ball blends between them.

Rolling ball blends are patches on *canal surfaces*, which are the envelopes of one-parameter families of spheres. The curve traced by the centres of the spheres is called its *spine curve*, and the varying radius is described by its *radius function*. It has been shown that a canal surface with rational spine and radius function has a rational parametrization [11], and an algorithm for minimal bi-degree $(n, 2)$ rational parametrizations of patches on canal surfaces was described in [7].

In [6] we specified the configurations of natural quadrics which admit rational fixed radius rolling ball blends for any radius R (for ease of notation we write *cone* for *right circular cone and/or cylinder*):

- A plane and a cone.
- Two cones with two points of oriented contact (see definition in Sect. 12.3).
- Two cones with one point of oriented contact.
- A sphere and a cone with one point of oriented contact.

We also presented closed expressions for their minimal bi-degree parametrizations.

The major restriction of fixed radius blends, apart from the limited range of configurations where such rational blends exist, is the limited flexibility inherent in their construction: their only degree of freedom is the radius of the blend. For the plane/cone blend in Fig. 12.1, this results in an uneven blend, where the width of

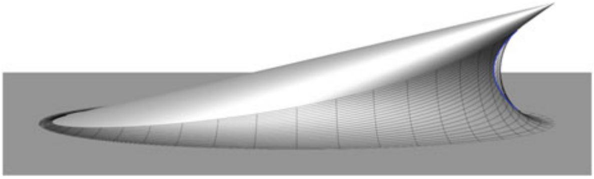


Fig. 12.1 Fixed radius rolling ball blend of a plane and a cone, with a local self-intersection on the right

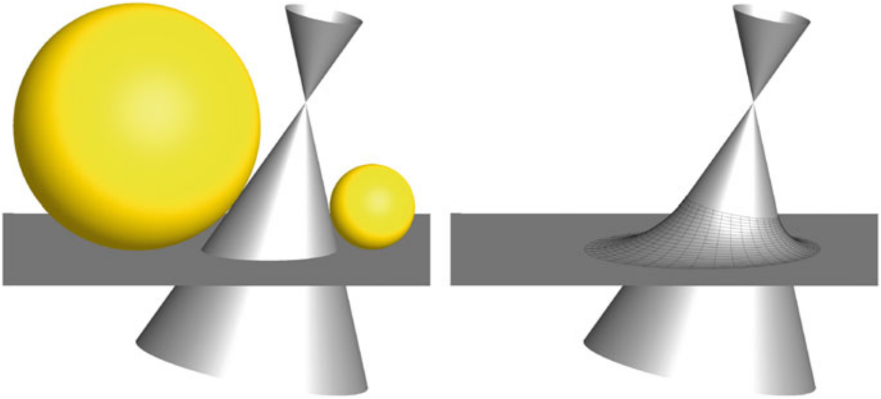


Fig. 12.2 A plane and a cone with two control spheres (*left*), and the resulting variable radius rolling ball blend (*right*)

the blend is much wider on the right than on the left. There is also an upper limit on the radius of the blend if we want to avoid local self-intersections in the blending surface, such as on the right hand side of the blend in Fig. 12.1.

When we compose several patches of blending surfaces to create a composite blend of a network of edges and corners, as is typically the case in the model of a physical object, the fixed radius is too severe a restriction when we want to ensure at least G^1 continuity between adjacent patches. By constructing variable radius rolling ball blends, we give designers added flexibility to adjust the blend for aesthetic and functional reasons.

In current CAD systems, the rolling ball blends that are implemented analytically, i.e., without approximation, are those that can be constructed from patches on spheres, cylinders and tori. Otherwise, the blend is typically created by specifying a curve in each surface and constructing a surface patch between them with a given level of continuity with the original surfaces.

We propose a more intuitive approach to the construction of variable radius rolling ball blends, using control spheres tangent to the original surfaces to specify the radius of the blend at certain key points. A simple illustration is given in Fig. 12.2, where two control spheres specify the radius of the blend above and below the cone, their centres contained in the plane of symmetry of the plane/cone

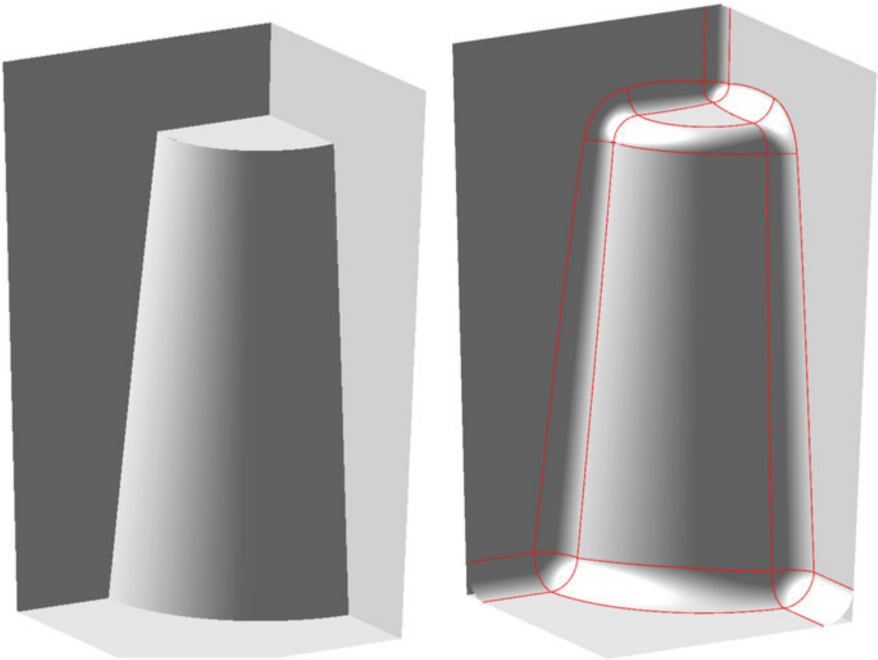


Fig. 12.3 Example of a composite corner. *Left*: A corner detail, consisting of a patch on a cone limited by four planes. *Right*: The composite blend, constructed in Sect. 12.5

configuration. If we require the blend to maintain the plane of symmetry, the two control spheres uniquely define a variable radius rolling ball blend with quadratic spine and radius function, shown on the right in the figure.

We will use the example corner in Fig. 12.3 to demonstrate how the control spheres are used to construct a patchwork of edge and corner blends with at least G^1 continuity between adjacent patches. The example corner consists of a patch on the cone

$$x^2 + y^2 = \frac{(z-26)^2}{4\sqrt{3}} \tag{12.1}$$

and four limiting planes $z = 0$ (bottom), $z = 10$ (top), $20\,y + z = 0$ (left), and $20\,x + z = 0$ (right). Such a corner detail may occur, e.g., when designing models for pressure molded plastic: in order to ensure that the component is easily removable from the mold, vertical surfaces are slightly tilted. This transforms a vertical cylindrical housing for a screw into a cone, and tilts the vertical planes slightly outwards. As the vertical planes no longer contain the axis of the cone, the cone/plane intersection cannot be blended by a cylindrical patch.

We will continue the example in Sect. 12.5. In the next section we will outline some of the theoretical background for our constructions in a brief introduction to Laguerre geometry, limiting the technical details to what we will need in the following sections. For further details about the subject see, e.g., [3, 5, 6, 9], or Peternell and Pottmann's paper [12]. Please note that the approach to Laguerre geometry in the last paper is slightly different than the one we are using, which is described in [5, 6, 9].

12.3 It's All Spheres: A Short Introduction to Laguerre Geometry

A fixed radius rolling ball blend of two surfaces is constructed by tracing the path of a sphere around their intersection in such a way that at any point the sphere is tangent to both surfaces. If we let the radius of the sphere vary as it moves along the two surfaces, this results in a variable radius rolling ball blend. By construction, rolling ball blends are G^1 continuous with the original surfaces.

Right circular cones and cylinders are linear canal surfaces: they are envelopes of families of spheres whose spine and radius functions are linear. Thus, when we consider rolling ball blends of the natural quadrics, all our objects can be expressed in terms of families of spheres.

In the description of rolling ball blends above, there is an element of ambiguity: in general, there is a choice of four placements of the rolling ball blend along an intersection of two surfaces. In order to have an unambiguous definition of the blend we assign each surface an orientation, given by the direction of its unit normal vector field. For a sphere, this is defined by the sign of its radius: if the radius is positive the unit normal vectors point towards its centre, if it is negative they point outwards. As envelope surfaces, canal surfaces inherit their orientation from their family of spheres. For a cone, this implies that the orientation of the two half cones on either side of its apex is opposite: the radius function is linear and passes through zero at the apex.

We say that two surfaces are in oriented contact at a point if they are tangent and their unit normal vectors coincide. By considering oriented surfaces, we can now define a rolling ball blend without the original ambiguity: it is the surface traced by an oriented sphere around the intersection of two oriented surfaces, in such a way that at any point the sphere is in oriented contact with both the original surfaces. The placement of the blend is then uniquely defined.

12.3.1 Minkowski Space

Since all our objects are defined by oriented spheres, we are interested in measuring the distance between two oriented spheres. Given two oriented spheres with centres $\vec{s}_0$ and $\vec{s}_1$, and radii r_0 and r_1, we represent them by the four-dimensional points

$\vec{p}_0 = \left(\vec{s}_0; r_0\right)$ and $\vec{p}_1 = \left(\vec{s}_1; r_1\right)$. The distance between $\vec{p}_0$ and $\vec{p}_1$ is given by the *Minkowski metric*

$$\left\| \vec{p}_1 - \vec{p}_0 \right\| = \sqrt{\left\langle \vec{p}_1 - \vec{p}_0, \vec{p}_1 - \vec{p}_0 \right\rangle} \tag{12.2}$$

where

$$\left\langle \vec{u}, \vec{v} \right\rangle = u_1 v_1 + u_2 v_2 + u_3 v_3 - u_4 v_4, \quad \vec{u} = (u_1, u_2, u_3; u_4), \quad \vec{v} = (v_1, v_2, v_3; v_4), \tag{12.3}$$

is the *Minkowski scalar product*. When $\left\langle \vec{p}_1 - \vec{p}_0, \vec{p}_1 - \vec{p}_0 \right\rangle \geq 0$, this is the tangential distance between the two spheres (see, e.g., [3], Fig. 3.6). Oriented contact between $\vec{p}_0$ and $\vec{p}_1$ then corresponds to $\left\| \vec{p}_1 - \vec{p}_0 \right\| = 0$.

The 4-dimensional space equipped with this scalar product is called *Minkowski space*, written $\mathbb{R}^{3,1}$, and can be interpreted as the space of all oriented spheres. A canal surface can then be represented as a curve in $\mathbb{R}^{3,1}$. The converse is not necessarily well-defined: in order to ensure that the envelope surface of the family of spheres is real, we require that the curve $\vec{f}(t) = \left(\vec{s}(t); r(t)\right) \in \mathbb{R}^{3,1}$ satisfies $\|\dot{\vec{f}}(t)\|^2 > 0$. While cones, as linear canal surfaces, satisfy this condition, it is worth noting that this is not the case for all lines in $\mathbb{R}^{3,1}$. Consider the line interpolating two points $\vec{p}_0$ and $\vec{p}_1$ corresponding to two spheres such that one is completely contained within the other. The derivative of this curve is proportional to $\vec{p}_1 - \vec{p}_0$, we have $\left\| \vec{p}_1 - \vec{p}_0 \right\|^2 < 0$, and the linear family of spheres does not have a real envelope.

In general we have three types of lines in $\mathbb{R}^{3,1}$ (see, e.g., [9]):

- *Hyperbolic lines*, corresponding to cones, where $\left\| \vec{p}_1 - \vec{p}_0 \right\|^2 > 0$;
- *Parabolic lines*, where $\left\| \vec{p}_1 - \vec{p}_0 \right\|^2 = 0$;
- *Elliptic lines*, where $\left\| \vec{p}_1 - \vec{p}_0 \right\|^2 < 0$.

Minkowski space is also used to represent the 4-dimensional space-time in which Einstein's theory of special relativity is formulated, so in the literature the types of lines are also sometimes called space-like, light-like and time-like, respectively.

A linear family of spheres corresponding to a line in $\mathbb{R}^{3,1}$ is called a *pencil of spheres*. Parabolic lines and the corresponding parabolic pencils of spheres are a useful tool in the construction of rolling ball blends. They are generated by two spheres in oriented contact, thus all spheres in the pencil are in oriented contact at this point. The point of oriented contact corresponds to the sphere with zero radius, and can thus be found by solving a linear equation. By considering the parabolic pencils generated by a sphere and a rolling ball, this gives us the *touching curve* on the sphere, i.e., the curve traced by the rolling ball along the sphere.

In order to determine the touching curve on a cone, we note that if the rolling ball is in oriented contact with the cone, then it is in oriented contact with exactly one sphere in the family associated with the cone. Thus the quadratic equation we get from the expression for the Minkowski metric in Eq. 12.2 is reduced to a linear equation for determining the sphere at zero distance from the rolling ball, i.e., we find the touching curve on a cone by solving two linear equations.

12.3.2 Curves in the Bisector in $\mathbb{R}^{3,1}$

When constructing a rolling ball blend, we want to determine a one-parameter family of spheres such that any sphere is in oriented contact with both surfaces. By construction, the spine curve of the corresponding canal surface lies in their bisector surface, so one approach to variable radius rolling ball blends is to consider rational curves in the bisector surface in $\mathbb{R}^3$. However, if our interest is in *rational* blend parametrizations, we need not only the spine curve but also the radius function to be rational. And in general, the distance of a rational curve from a cone is not rational.

Consider the family of all spheres in oriented contact with a given surface. This corresponds to a hypersurface in $\mathbb{R}^{3,1}$, called its *isotropic hypersurface*. By intersecting two isotropic hypersurfaces, we get the 2-dimensional surface in $\mathbb{R}^{3,1}$ corresponding to all spheres in oriented contact with both original surfaces. This is called their *bisector surface in* $\mathbb{R}^{3,1}$. Any rational curve in this bisector surface corresponds to a canal surface with rational spine curve and radius function, and thus to a rational rolling ball blend of the two original surfaces.

Given an oriented plane Π in $\mathbb{R}^3$:

$$\Pi : Ax_1 + Bx_2 + Cx_3 + D = 0, \quad A^2 + B^2 + C^2 = 1, \tag{12.4}$$

with unit normal vector $(A, B, C)^T$, the spheres in oriented contact with Π define a hyperplane Π_m in $\mathbb{R}^{3,1}$:

$$\Pi_m : Ax_1 + Bx_2 + Cx_3 - x_4 + D = 0. \tag{12.5}$$

The isotropic hypersurface of a cone is its *isotropic quadric* (see [6]):

$$\left\langle \vec{x} - \vec{p}_0, \vec{p}_1 - \vec{p}_0 \right\rangle^2 = \left\| \vec{x} - \vec{p}_0 \right\|^2 \left\| \vec{p}_1 - \vec{p}_0 \right\|^2 \tag{12.6}$$

where $\vec{p}_0$ and $\vec{p}_1$ are two distinct points on the hyperbolic line corresponding to the cone, and $\vec{x} = (x_1, x_2, x_3; x_4)$.

In general, the problem of constructing a rational rolling ball blend between two surfaces can be reduced to the problem of constructing a rational curve in their bisector surface in $\mathbb{R}^{3,1}$, then applying the algorithms in [5,6]. When considering two natural quadrics, their bisector surface has a rational parametrization. We can thus generate rational curves in the bisector from rational planar curves in the parameter domain. However, the parametrization degree of these curves is necessarily high. For the purpose of parametrizing rolling ball blends, we want relatively low-degree curves in $\mathbb{R}^{3,1}$, in order to keep the bi-degree of the resulting surface parametrization as low as possible. We are thus primarily interested in constructing rational curves of low degree in the bisector surface in $\mathbb{R}^{3,1}$.

If an approximate parametrization of the blending surface is sufficient, we can apply the results in [8] to construct a spline curve in the bisector surface, giving us a large degree of flexibility in the construction. For exact parametrizations we need to consider rational curves in 2-dimensional surfaces in $\mathbb{R}^{3,1}$. A first approach is to consider hyperplane sections of the bisector surface in $\mathbb{R}^{3,1}$. This is a direct generalization of the construction of blends of fixed radius R, where the curve is determined by the intersection with the hyperplane $x_4 = R$. However, even when a bisector surface is rational, this is not necessarily the case for its hyperplane sections. There are only two types of surfaces where any hyperplane section is rational: rational ruled surfaces and the quartic Steiner surface [10]. This gives us the configurations with rational fixed radius blends at the beginning of Sect. 12.2:

- A plane and a cone.
- Two cones with two points of oriented contact.
- Two cones with one point of oriented contact.
- A sphere and a cone with one point of oriented contact.

In the first two cases the bisector surface in $\mathbb{R}^{3,1}$ is a rational ruled quadric, in the last two it is a quartic Steiner surface. The hyperplane sections of the bisector surfaces in $\mathbb{R}^{3,1}$ are then one or two conics (first and second case), or a singular quartic curve (third and fourth case). Note that though the curve in $\mathbb{R}^{3,1}$ is contained in a hyperplane section, the spine curve of the corresponding canal surface is not necessarily planar (see Fig. 12.5, left); it only means that its radius function depends linearly on the coordinates of the spine curve.

To summarize, planes, natural quadrics, and rolling ball blends between them can all be represented as hyperplanes, points, and curves in $\mathbb{R}^{3,1}$. By parametrizing rational curves in the bisector surface in $\mathbb{R}^{3,1}$, we can construct rational variable radius rolling ball blends of the original surfaces.

In the next section we will outline how a composite corner blend can be constructed using mainly hyperplane sections of individual edge blends, while maintaining at least G^1 continuity between adjacent patches. We will then apply this to construct a blend of the composite corner in Fig. 12.3.

12.4 Variable Radius Rolling Ball Blends

We propose using a set of carefully chosen control spheres to construct variable radius rolling ball blends of composite corners. The placement of the control spheres is guided by the intended sequence of construction of the blends, and by the level of continuity required between adjacent patches. From the control spheres we construct curve segments in $\mathbb{R}^{3,1}$ corresponding to variable rolling ball blends of edges and corners. Using the parametrization algorithm in [5], we can then construct the rational parametrizations of the individual blending patches.

Before we outline the algorithm for the construction of composite blends, we need to describe how individual edge and corner blend patches are constructed.

12.4.1 Edge Blends

In Sect. 12.3 we described how a canal surface can be represented as a curve in $\mathbb{R}^{3,1}$. A segment of a canal surface containing an edge blend can therefore be constructed by specifying the two end spheres and a curve segment connecting them, in the bisector surface in $\mathbb{R}^{3,1}$. In this paper we use two approaches for constructing such curve segments: (rational) Bézier curves in planar bisector surfaces, and hyperplane sections of non-planar bisector surfaces. In general, any technique for constructing rational curves between two points in a 2-dimensional surface can be applied, taking into account the restrictions on tangent directions at the joins of two adjacent curve segments.

The bisector surface in $\mathbb{R}^{3,1}$ of two oriented planes in $\mathbb{R}^3$ is a 2-dimensional plane in $\mathbb{R}^{3,1}$, and any rational curve in this plane corresponds to a rational blend of the two planes in $\mathbb{R}^3$. Representing the curve in Bézier form, its control points correspond to control spheres in oriented contact with both the original planes. This is shown on the right in Fig. 12.4.

When the bisector surface is non-planar, we can construct the curve segment by taking the intersection of the bisector surface with a hyperplane containing the points corresponding to the two end spheres. In general we need four points to define a hyperplane in $\mathbb{R}^{3,1}$, but when intersecting the bisector in $\mathbb{R}^{3,1}$ of a cone and a plane Π the intersection of a hyperplane with Π_m is a 2-dimensional plane. The implicit equation for Π_m express x_4 in terms of the other variables (its coefficient is non-zero), thus we may eliminate x_4 from the expression of the intersecting hyperplane

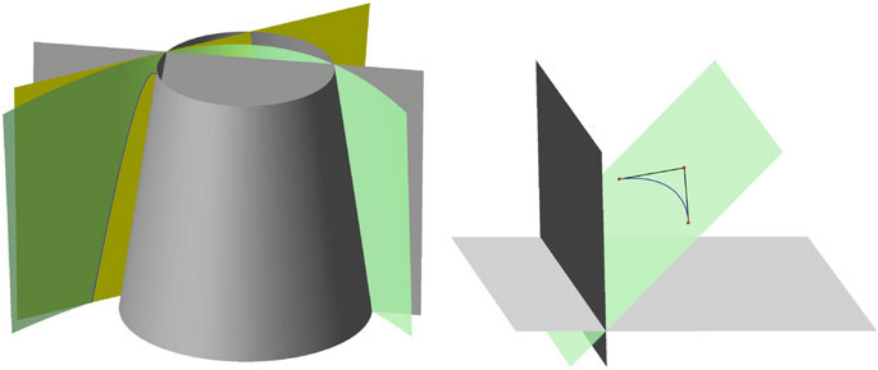


Fig. 12.4 Curves in the bisector surface in $\mathbb{R}^{3,1}$, projected to their first three coordinates, i.e., the spine curve of the corresponding canal surface in the bisector surface in $\mathbb{R}^3$ (*green*). *Left*: Intersection with a 2-dimensional plane (*yellow*). *Right*: A Bézier curve in the bisector plane

reducing it to a linear equation in three variables. It is therefore sufficient to choose three points in the bisector surface in $\mathbb{R}^{3,1}$ to uniquely define the intersection with the isotropic quadric of the cone. Given the two end spheres, this leaves us one degree of freedom in the choice of the hyperplane section.

In this case we can visualize the hyperplane section directly in $\mathbb{R}^3$. The centres of the three control spheres are points on the bisector surface of the cone and the plane Π in $\mathbb{R}^3$, and they define a plane intersecting the bisector surface in the quadratic spine curve of the blend. As the distance from a rational curve to a plane is rational, we can thus do the construction directly in $\mathbb{R}^3$ in this case. This approach is shown on the left in Fig. 12.4. The spine curves shown in this figure correspond to the blends of the left plane with the cone, and the top plane with the left plane, in the example composite corner in Fig. 12.3. The parametrized blends are shown on the right in Fig. 12.3.

When constructing variable radius blends of two cones, however, a hyperplane section in $\mathbb{R}^{3,1}$ cannot be reduced to the construction of a 2-dimensional plane, and thus its visualization in $\mathbb{R}^3$ is not as straightforward as in the plane/cone case. The hyperplane gives us a linear relation

$$Ax_1 + Bx_2 + Cx_3 + Ex_4 + D = 0 \tag{12.7}$$

which can be interpreted as assigning a radius x_4 to each point $(x_1, x_2, x_3) \in \mathbb{R}^3$. We can visualize this by considering the plane $Ax_1 + Bx_2 + Cx_3 + D = 0$ in $\mathbb{R}^3$ where the assigned radius is zero. As we move away from this plane in the normal direction, the radii increase linearly at a rate of change of $E/(A^2 + B^2 + C^2)$.

The choice of a hyperplane can thus be visualized as the choice of a unit vector (A, B, C) giving the direction of the maximum rate of change in radius, the rate of change E of the radius in this direction, and the distance D of the zero-radius plane from the origin. This is illustrated in Fig. 12.5, where we construct a variable radius blend of two perpendicular cylinders in oriented contact at one point, with radius 1 and 2 respectively (recall that we need at least one point of oriented contact to ensure that any hyperplane section is rational). We choose as (A, B, C) the common unit normal vector at the point of oriented contact, so that the symmetry of the configuration of the two cylinders is maintained in the blend. The blending radius below the smallest cylinder is fixed by placing a control sphere, thus determining D. By varying E we obtain a series of hyperplane sections of the bisector surface in $\mathbb{R}^{3,1}$, illustrated on the left of Fig. 12.5. Note that all the spine curves pass through a common point below the smaller cylinder, at the centre of the control sphere. On the right of the figure, we show the narrowest of the blends, and the limiting curves of the blends as E changes.

12.4.2 Corner Blends

We distinguish between two types of three-sided corners: *homogeneous corners* where all the adjoining edges are convex/concave, such as the central top corner and

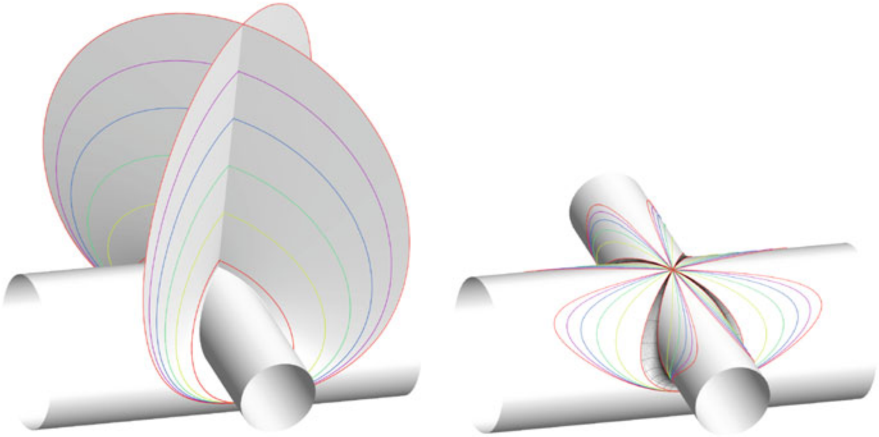


Fig. 12.5 Symmetric variable radius rolling ball blends of two cylinders with one point of oriented contact. *Left*: The spine curves in the bisector in $\mathbb{R}^3$, projected from hyperplane sections of the bisector in $\mathbb{R}^{3,1}$. *Right*: A variable radius blend of the two cylinders, and the limiting curves of the blends corresponding to the spine curves on the *left*

the two bottom corners in Fig. 12.3, and *heterogeneous corners* where one of the edges differs from the two others, such as the corners at the ends of the intersection of the cone with the top plane.

At homogeneous corners we can connect the edge blends by simply requiring that they all end at a sphere tangent to all three faces. Two adjacent edge blends will then meet in the touching point of the sphere with their common face. The corner patch is a patch on the sphere, limited by three circular arcs, and is by construction G^1 continuous with the edge blends. By considering the edge blends as Bézier curves of spheres, we can also specify the necessary conditions for achieving G^2 continuity with the corner sphere: it has to correspond to a double control point (see [5], Rem. 11). This approach can be generalized to n-sided corners, as long as there exists a sphere tangent to all faces, which is equivalent to requiring that the corner circumscribes a right circular cone.

In order to ensure at least G^1 continuity, we must make sure that adjacent edge blends do not overlap, i.e., that two adjacent arcs of circles limiting the spherical patch only intersect at the touching point. For a three-sided corner, this condition is described in Fig. 9 of [5]: in the control polygon of the spine curve, the next to last control point must avoid the triangle defined by the vertex of the corner, the centre of the corner sphere, and the point on the edge $2R$ distant from the vertex, where R is the radius of the corner sphere. A similar condition can be derived for n-sided corners.

At heterogeneous corners we cannot find a sphere tangent to all faces, so we need an alternative approach. When the face opposite the single convex/concave edge is planar, we can use a patch on a Dupin cyclide to blend the corner. This approach is also valid for homogeneous corners.

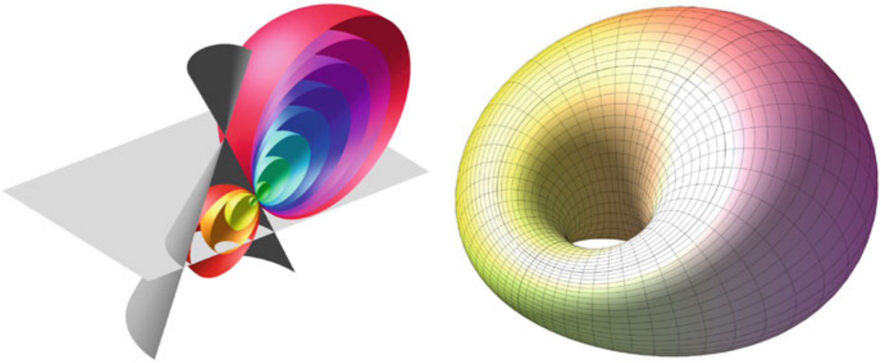


Fig. 12.6 A cone with a parabolic pencil of spheres (*left*), and the Dupin cyclide containing the blend with the plane (*right*). The cone, plane and blend are shown on the *right* in Fig. 12.2

The construction makes use of the following theorem:

Theorem 12.1 *Given an oriented canal surface and an oriented plane, for each characteristic circle on the canal surface there exists a unique oriented Dupin cyclide containing a patch blending the canal surface with the plane along the characteristic circle.*

Proof The canal surface is in oriented contact with one of the spheres in its one-parameter family along a *characteristic circle*. We will prove the theorem by showing that given a plane Π and a circle on a sphere Σ, the one-parameter family of spheres in oriented contact with both the sphere along the circle, and the plane, corresponds to a pseudo-Euclidean (PE) circle in $\mathbb{R}^{3,1}$, i.e., its envelope is a Dupin cyclide (Fig. 12.6, right).

A point $\vec{p}(t)$ on the circle is identified with the zero radius sphere $(\vec{p}(t); 0)$, which together with Σ generate a parabolic line in $\mathbb{R}^{3,1}$. The union of the parabolic lines through points on the circle is a quadratic cone, whose infinite points are all contained in the absolute quadric Ω:

$$\Omega : x_0 = 0, \quad x_1^2 + x_2^2 + x_3^2 - x_4^2 = 0. \tag{12.8}$$

Here $(x_0, x_1, x_2, x_3; x_4) \in \mathbb{P}^4$ is a point in projective space corresponding to the affine point $\left(\frac{x_1}{x_0}, \frac{x_2}{x_0}, \frac{x_3}{x_0}; \frac{x_4}{x_0}\right) \in \mathbb{R}^{3,1}$.

The intersection of the cone with the hyperplane Π_m, i.e., the sphere in each parabolic pencil in oriented contact with Π, is therefore a conic with two infinite points on Ω, which is one definition of a PE circle. This proves the theorem. □

Remark 12.1 The Dupin cyclide blend of a cone and a plane can also be found in [1, 2, 13].

In a composite corner, a heterogeneous corner blend can be constructed by first blending the single concave/convex edge. This determines the characteristic circle on the end sphere of this edge blend. Applying the procedure described in the proof above, i.e., intersecting the quadratic cone with the hyperplane, gives a rational parametrization of the PE circle corresponding to the cyclide blend of the end sphere, and thus the edge blend, with the plane.

At either end of the cyclide blend, the tangent lines of the PE circle determine the tangent cones of the adjacent edge blends. Similarly, we can specify one of the end spheres of the cyclide corner blend and its tangent cone, to uniquely determine the end spheres and tangent cones of the two remaining edge blends adjacent to the corner. Dupin cyclide corner blends therefore have one primary edge, whose blend is constructed before the two others. As opposed to spherical corner blends they influence the order in which the individual blending patches are constructed.

12.4.3 Constructing Composite Variable Radius Rolling Ball Blends

In the case of a single edge blend, the only restriction on our choice of control spheres is that they should be in oriented contact with both surfaces, i.e., that the corresponding points $\mathbb{R}^{3,1}$ lie in the bisector surface. When constructing a composite blend of a network of edges and corners, we also need to consider the continuity between adjacent blending patches. In [5] we give the conditions for when the join of two canal surfaces is G^1 and G^2 continuous: it is equivalent to G^1 and G^2 continuity of the join between the corresponding curves in $\mathbb{R}^{3,1}$. The G^2 condition is too strict to expect to be able to construct a composite blending patch with internal G^2 continuity using mainly hyperplane sections of the bisector surface in $\mathbb{R}^{3,1}$, so in the following we will focus on the construction of a composite blend with internal G^1 continuity.

The condition for G^1 continuity means that the curve segments in $\mathbb{R}^{3,1}$ corresponding to two adjacent blending patches are joined in a point where their tangent lines coincide. The envelope in $\mathbb{R}^3$ of the tangent line of a curve in $\mathbb{R}^{3,1}$ is called the *tangent cone* of the canal surface, and is by construction G^1 continuous with the canal surface along the characteristic circle.

In the sequential construction of a composite blend, one end sphere and tangent cone is given by the preceding blending patch. A hyperplane is then defined by the choice of the second end sphere, and if there are sufficient degrees of freedom, a final control point or tangent cone. In general, any curve in the bisector surface in $\mathbb{R}^{3,1}$ connecting the two endpoints with the required initial tangent line will produce an edge blend of G^1 continuity with the preceding blending patch.

Remark 12.2 At the end of a curve segment, when we use its tangent line to construct the hyperplane section for the adjacent curve, we ensure that the two curves are G^1 continuous at the join. However, as we are considering only a segment of the second curve, the resulting composite curve may have a singularity at the join (Fig. 12.7, left, where the second end sphere lies on the left hand segment of the

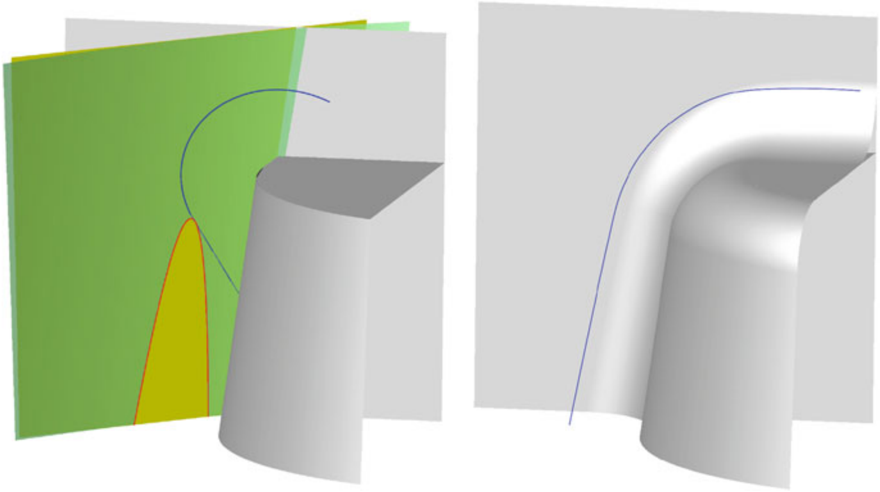


Fig. 12.7 *Left*: The spine curve (*red*) of the blend of the cone with the *left plane*, is the intersection of a plane (*yellow*) with the bisector surface (*green*). The intersection curve is tangent to the spine curve (*blue*) of the adjacent corner blend, but the join of the *left red* segment with the *top blue* segment is singular. *Right*: The spine curve and blending surface of a non-singular G^1 join

intersection curve). When we make sure that the join is non-singular, both the join of the curve segments in $\mathbb{R}^{3,1}$ and the join of the blending surfaces in $\mathbb{R}^3$ are G^1 (Fig. 12.7, right).

We start the construction of the composite blend by identifying the heterogeneous corners, checking that the face opposite the single concave/convex edge is planar. We then choose which of the homogeneous corners will also be blended with Dupin cyclide patches. These cyclide corner blends will determine the order in which the individual blending patches of the composite corner will be constructed. Care must be taken with edges where the corner blends at both ends are cyclide patches, to ensure that there are enough degrees of freedom to construct the edge blend. Such edges are therefore natural starting points in the sequence of blend constructions.

After choosing the primary edge for each cyclide corner blend, we choose the order in which the blending patches will be constructed, making sure that in each case the primary edge is blended first. We then place the control spheres and tangent cones which will determine the hyperplane sections and thus the curve segments in $\mathbb{R}^{3,1}$ corresponding to the blending patches.

We start by placing control spheres, tangent to all three faces, at the homogeneous corners which will be blended by spherical patches. In addition to specifying the blending radii at the end of the adjacent edge blends, these control spheres contain the spherical corner blends.

We then place a control sphere in oriented contact with the two faces of the primary edge of each Dupin cyclide corner, specifying the blending radius at the end of the primary edge blend. This control sphere uniquely determines two secondary control spheres at the end of the two remaining edge blends adjacent to the corner: they are in oriented contact with the control sphere at its points of oriented contact with the two faces, and with the face opposite the primary edge. The tangent cone at the end of the primary edge blend then uniquely determines the Dupin cyclide corner blend. At the two secondary control spheres, the cyclide patch determines the tangent cones of the adjacent edge blends.

These control spheres define the blending radii at the corners of the composite blend. It remains to place additional control spheres where needed in order to define the curve segments corresponding to edge blends, either for hyperplane sections of the bisector surface in $\mathbb{R}^{3,1}$, or as control points of a Bézier curve in a planar bisector surface. Note that this last type of control spheres are slightly different from the others: apart from the endpoints they do not lie on the curve segment in $\mathbb{R}^{3,1}$.

After completing the placement of the control spheres we follow the methods outlined above, and sequentially construct curve segments in $\mathbb{R}^{3,1}$ corresponding to blends of the edges and corners of the composite corner. Using the parametrization algorithm in [5], we then construct the rational parametrization of the blends.

The following algorithm summarises the above construction:

Algorithm 12.1 *A rational variable radius rolling ball blend of a composite corner can be constructed with the following algorithm as long as four conditions are satisfied:*

- *All faces are patches on planes or on natural quadrics.*
- *Two quadric patches meeting along an edge are patches on two oriented quadrics with one or two points of oriented contact.*
- *All corners have three adjoining edges.*
- *At all heterogeneous corners, the face opposing the single convex/concave edge is planar.*

The blend is then constructed by completing the following steps:

1. *For each heterogeneous corner, verify that the face opposite the single convex/concave edge is planar. These corners will all be blended by patches on Dupin cyclides.*
2. *Choose which homogeneous corners will be blended by spherical patches, and which by Dupin cyclide patches.*
3. *For each Dupin cyclide corner choose the primary edge, making sure that the opposite face is planar.*
4. *Determine the sequence in which the individual blending patches will be constructed, respecting the order imposed by the Dupin cyclide corners.*
5. *Place a control sphere at each spherical corner, tangent to all three faces.*
6. *Place a control sphere at the primary edge of each Dupin cyclide corner.*
7. *Derive the secondary control spheres at each Dupin cyclide corner.*
8. *Determine the control spheres where a tangent cone will be determined by the adjacent blending patch.*
9. *Place additional control spheres and/or tangent cones as needed for the edge blends, to properly define hyperplanes and/or control points for Bézier curves.*

10. *Sequentially parametrize the curve segments in* $\mathbb{R}^{3,1}$.
11. *Parametrize the blending patches by applying Alg. 2 of [5].*

This algorithm can easily be extended to include homogeneous corners that circumscribe cones, but for the sake of clarity we chose to formulate Algorithm 12.1 for three-sided corners only. The second condition of the algorithm can be relaxed if we construct the edge blends directly from rational curves in the bisector in $\mathbb{R}^{3,1}$: if there is no oriented contact this surface is still rational, though we cannot guarantee the rationality of its hyperplane sections.

The fourth condition is sufficient to guarantee that we can construct a Dupin cyclide corner blend at the end of the edge blend. This may still be possible if the face opposing the single convex/concave edge is a patch on a cone. In fact, this is possible if and only if the cone and the tangent cone at the end of the opposite edge blend inscribe a common sphere, i.e., are in oriented contact at two points. In that case the two corresponding lines in $\mathbb{R}^{3,1}$ intersect, spanning a 2-dimensional plane. A PE circle is uniquely defined by three finite points on it (see [9], Rem. 8), thus these two lines, and the point corresponding to the end sphere at the tangent cone, define a PE circle and thus a Dupin cyclide blend.

Finally, the approach in Algorithm 12.1 can be extended to other types of surfaces, and PN surfaces in general: see Sect. 12.6.

12.5 Blending the Example Corner

To demonstrate Algorithm 12.1, we apply it to the construction of a blend of the composite corner in Fig. 12.3. The example has been constructed to satisfy the four conditions for applying the algorithm.

1. *Heterogeneous corners*: The composite corner has two heterogeneous corners, one at each end of the intersection of the cone with the top plane (the top arc). In each case, the opposite face is planar: the left and right plane.
2. *Homogeneous corners*: For this example, we choose to blend all the homogeneous corners with spherical patches.
3. *Primary edges*: For the two heterogeneous corners, the primary edge is the top arc.
4. *Sequence of construction*: As recommended above, we start the blending construction at the top arc, to ensure that we have enough degrees of freedom to construct the hyperplane section. We then blend the heterogeneous corners at each end of the top arc, before we blend the left and right plane/cone intersections (the vertical blends), and the intersections of the top plane with the left and right plane (the top blends). The remaining edges can be constructed in any order, e.g., starting with the intersection of the cone with the bottom plane (the bottom arc) before blending the remaining plane/plane intersections.

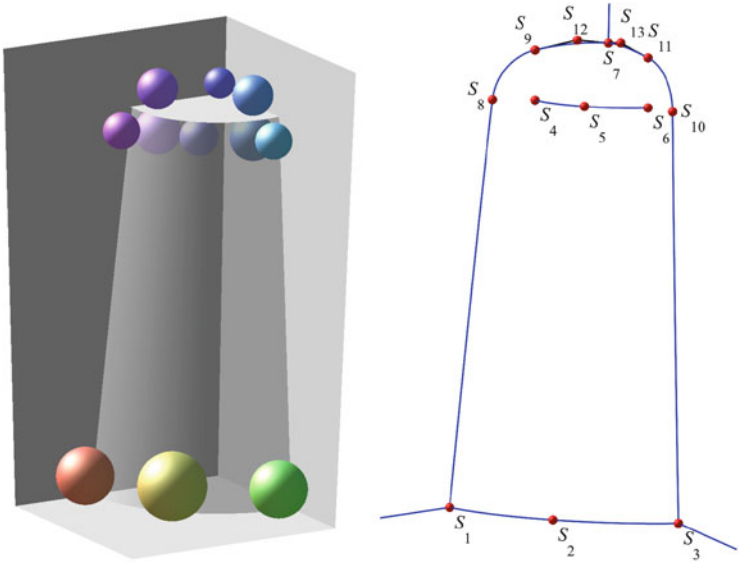


Fig. 12.8 The control spheres of the composite blend (S_{12} and S_{13} have been omitted on the *left*). The graph on the *right* shows the centres of the control spheres, and the segments of spine curves we construct between them

5. *Spherical corners*: We place control spheres at the spherical corners, S_1, S_3, and S_7 in Fig. 12.8.
6. *Dupin cyclide corners*: At the Dupin cyclide corners, we place the control spheres S_4 and S_6 as the end spheres of the blend of the top arc.
7. *Secondary control spheres*: The placement of these spheres determine the secondary control spheres S_8, S_9, S_{10}, and S_{11}.
8. *Tangent cones*: The blend of the top arc determines the tangent cones at each end, and the Dupin cyclide blends at each end of the top arc determine the tangent cones at the top of the vertical blends, and at one end of each top blend.
9. *Remaining control spheres/tangent cones*: For the top arc we need an additional control sphere to uniquely define the 2-dimensional plane in $\mathbb{R}^{3,1}$, so we choose S_5 at its midpoint. This is also the case for the bottom arc, so we choose S_2 at its midpoint. The vertical blends are completely defined by two end spheres and one tangent cone each. The top blends will be constructed as Bézier curves in the bisector plane in $\mathbb{R}^{3,1}$. Their end spheres are already defined, and the tangent direction at the heterogeneous corners given. We choose a middle control sphere for each blend, S_{12} and S_{13}, along these tangent directions. In addition, we choose that the continuity with the spherical blend at S_7 should be G^2, and so this control sphere should be a double control point for both Bézier curves. The remaining plane/plane blends will be patches on cylinders, so no further control spheres/tangent cones are needed.

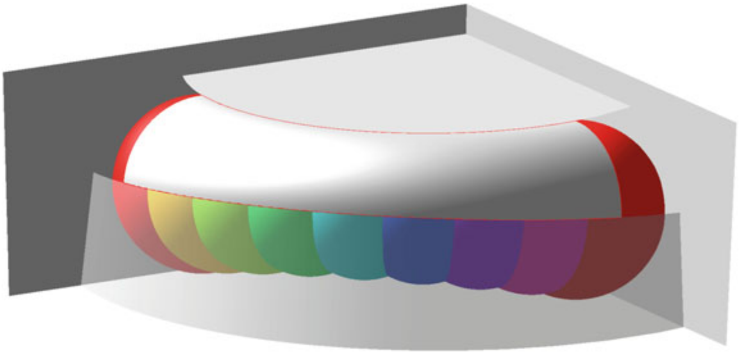


Fig. 12.9 Blending the intersection of the cone with the *top plane*: The one-parameter family of spheres and the blending patch

10. *Curve segments in* $\mathbb{R}^{3,1}$:
    - The three control spheres S_4, S_5, and S_6 define a 2-dimensional plane in $\mathbb{R}^{3,1}$. Its intersection with the isotropic quadric of the cone is a conic. The corresponding one-parameter family of spheres and the resulting blending patch are shown in Fig. 12.9.
    - The tangent lines at each side of this conic determine the PE circles corresponding to the two Dupin cyclide blends of the heterogeneous corners, which in turn determine the tangent lines at the secondary control spheres S_8, S_9, S_{10}, and S_{11}. The left cyclide blend, with its one-parameter family of spheres, and two tangent cones, is shown in Fig. 12.10.
    - For the vertical blends, S_1, S_8, and the tangent line at S_8, and S_3, S_{10}, and the tangent line at S_{10}, define 2-dimensional planes in $\mathbb{R}^{3,1}$, whose intersections with the isotropic quadric of the cone are conics. The spine curve of the left vertical blend is shown on the left in Fig. 12.4.
    - For the bottom arc, the three control spheres S_1, S_2, and S_3 define a 2-dimensional plane in $\mathbb{R}^{3,1}$, whose intersection with the isotropic quadric of the cone is a conic.
    - For the top blends, the control spheres $\{S_9, S_{12}, S_7, S_7\}$ and $\{S_{11}, S_{13}, S_7, S_7\}$ define the control polygons of cubic Bézier curves with the required continuity at each end. The spine curve and control polygon of the top left blend is shown on the right in Fig. 12.4.
    - As patches on cylinders, the remaining plane/plane blends are parametrized as line segments in $\mathbb{R}^{3,1}$.
11. Finally, we parametrize the individual blending patches by applying Alg. 2 of [5]. The complete composite blending surface is shown on the right in Fig. 12.3.

This completes the composite blend, which has internal G^1 continuity at all joins, except for the internal joins at S_7, which are G^2 continuous.

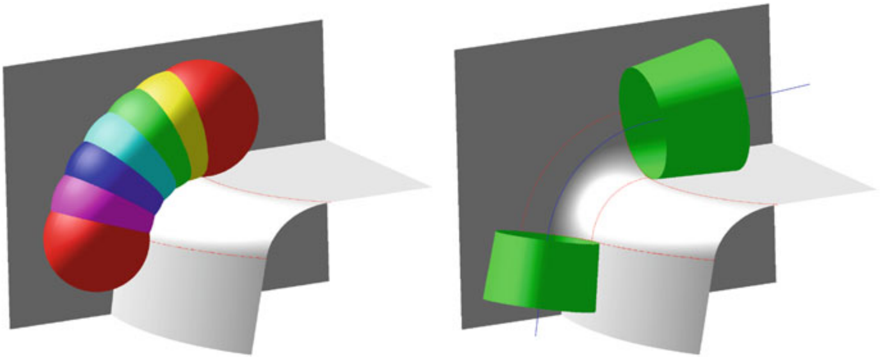


Fig. 12.10 Blending the *left* non-convex corner. *Left*: The one-parameter family of spheres in oriented contact with both the *left plane* and the blend of the top arc. *Right*: The cyclide corner blend with its tangent cones

In this construction, the control spheres are chosen with different degrees of freedom. At the spherical corners, the choice of radius determines the control spheres, so S_1, S_3, and S_7 are chosen with one degree of freedom. At the Dupin cyclide corners, any point in the bisector surface of the cone and the top plane can be used as endpoints, so S_4 and S_6 are chosen with two degrees of freedom. In practice, however, we want to use them to specify the radius of the blend at each end of the top arc, so their placement is restricted to the areas close to the Dupin cyclide corners. The secondary control spheres S_8, S_9, S_{10}, and S_{11} are determined by the choice of S_4 and S_6, and as such contribute no degrees of freedom to the construction. At the midpoints of the two arcs, S_2 and S_5 are chosen with one degree of freedom, as the radius here determines the 2-dimensional plane used to construct the curve segment. As S_{12} and S_{13} are chosen along the tangent lines of the left and right PE circle respectively, they are also chosen with one degree of freedom.

All in all, we have seven control spheres with one degree of freedom (S_1, S_2, S_3, S_5, S_7, S_{12}, and S_{13}), two with two degrees of freedom (S_4 and S_6), and four with none (S_8, S_9, S_{10}, and S_{11}). These control spheres completely determine the composite blend, and give us a total of 11 degrees of freedom for this composite corner.

Remark 12.3 This is, however, not the only way to construct the blend. We chose the order in which the blends are constructed, and chose to use cubic Bézier curves.

For the blends in the figures, we have chosen $r_1 = 0.75$, $r_2 = 0.9$, $r_3 = 0.75$, $r_4 = -0.6$, $r_5 = -0.5$, $r_6 = -0.6$, and $r_7 = 0.4$. At S_4 and S_6, we also choose $y_4 = 0$ and $x_6 = 0$, respectively. We choose $S_{12} = (0.80, 0.11, 10.64; 0.64)^T$ and $S_{13} = (0.11, 0.80, 10.64; 0.64)^T$ along the tangent lines at S_9 and S_{11}. Note that the signs of the radii determine whether the control spheres are in front of or behind the surface patches: S_4, S_5, and S_6 all have negative radii, and are therefore placed behind the blend of the top arc.

If Algorithm 12.1 were to be implemented in, e.g., a commercial CAD system, we envision that the system would present an initial composite blend with a limited variation in blend radius. The control spheres could then be displayed and directly manipulated to adjust the blend, both in terms of the variation in radius and the level of continuity with the spherical patches.

12.6 Generalizing the Approach to Blends of PN Surfaces

The isotropic quadric of a cone can be generalized to oriented surfaces with a well defined unit normal vector field. Consider a surface parametrized by $\vec{f}(s,t)$, with unit normal vector field $\vec{n}(s,t)$. A point on the surface can be considered as a sphere with zero radius, and the sphere $\vec{F}(r,s,t) = (\vec{f}(s,t) + r\,\vec{n}(s,t); r)$ is in oriented contact with the surface at $\vec{f}(s,t)$. In fact, $\vec{F}(r,s,t) \in \mathbb{R}^{3,1}$ is a parametrization of the isotropic hypersurface of $\vec{f}(s,t)$. For a PN surface, where both $\vec{f}(s,t)$ and $\vec{n}(s,t)$ are rational, the isotropic hypersurface is rational as well.

To parametrize variable radius rolling ball blends between two PN surfaces, we then need to consider curves in the intersection of two such isotropic hypersurfaces. In general, the intersection of two such hypersurfaces is not necessarily rational, and even when it is, there may not be many families of rational curves of relatively low degree on it. A starting point for further research would be to investigate rational variable radius rolling ball blends of PN surfaces, i.e., rational curves in certain 2-dimensional surfaces in $\mathbb{R}^{3,1}$. Blending a plane and a PN surface may be a natural starting point: the intersection of each parabolic pencil with the hyperplane is rational, which gives us a rational parametrization of the bisector surface in $\mathbb{R}^{3,1}$.

Conclusions

We have outlined an algorithm for constructing variable radius rolling ball blends of composite corners, with internal G^1 continuity between adjacent blending patches. The basis for the algorithm is the use of control spheres to specify the blending radius at certain key points. The approach was illustrated by applying the algorithm to the example corner in Fig. 12.3.

When the composite blend is rational, it is a PN surface, with an immediate extension to rational parametrizations of thick surfaces. As relatively simple, yet versatile volume parametrizations, we expect this class of rational volume representations to be useful in Isogeometric Analysis.

There are several possible avenues of future research:

- The construction of rational curves in 2-dimensional surfaces in $\mathbb{R}^{3,1}$, in order to extend the class of rational variable radius rolling ball blends.

(continued)

- More general blending surfaces, to extend the configurations of natural quadrics with rational blends.
- Construction of blends between a larger class of primitive surfaces, such as PN surfaces.

Acknowledgements We would like to thank Rimvydas Krasauskas for his improvement of our original proof of Theorem 12.1, and the editor and reviewers for their constructive feedback.

References

1. S. Allen, D. Dutta, Cyclides in pure blending I. Comput. Aided Geom. Des. **14**(1), 51–75 (1997)
2. S. Allen, D. Dutta, Cyclides in pure blending II. Comput. Aided Geom. Des. **14**(1), 77–102 (1997)
3. T.E. Cecil, *Lie Sphere Geometry* (Springer, New York, 2008)
4. J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA* (Wiley, Chichester/Hoboken, 2009)
5. H.E.I. Dahl, Piecewise rational parametrizations of canal surfaces, in *Mathematical Methods for Curves and Surfaces*. Lecture Notes in Computer Science, vol. 8177 (Springer, Berlin/New York, 2014), pp. 88–111
6. H.E.I. Dahl, R. Krasauskas, Rational fixed radius rolling ball blends between natural quadrics. Comput. Aided Geom. Des. **29**, 691–706 (2012)
7. R. Krasauskas, Minimal rational parametrizations of canal surfaces. Computing **79**(2–4), 281–290 (2007)
8. R. Krasauskas, M. Kazakevičit, Universal rational parametrizations and spline curves on toric surfaces, in *Computational Methods for Algebraic Spline Surfaces*, ed. by T. Dokken, B. Jüttler (Springer, Berlin/New York, 2005), pp. 213–231
9. R. Krasauskas, C. Mäurer, Studying cyclides with Laguerre geometry. Comput. Aided Geom. Des. **17**(2), 101–126 (2000)
10. E.H. Moore, Algebraic surfaces of which every plane-section is unicursal in the light of n-dimensional geometry. Am. J. Math. **10**(1), 17–28 (1887)
11. M. Peternell, H. Pottmann, Computing rational parametrizations of canal surfaces. J. Symb. Comput. **23**(2–3), 255–266 (1997)
12. M. Peternell, H. Pottmann, Applications of Laguerre geometry in CAGD. Comput. Aided Geom. Des. **15**(2), 165–186 (1998)
13. M. Pratt, Applications of cyclide surfaces in geometric modelling, in *3rd IMA Conference on the Mathematics of Surfaces*, Oxford, 1988

Part IV
Practical Industrial Problems

Chapter 13
Bisectors and Voronoï Diagram of a Family of Parallel Half-Lines

I. Adamou, M. Fioravanti, L. Gonzalez-Vega, and B. Mourrain

13.1 Introduction

Computer aided design (CAD) technology is currently facing limitations mainly due to conflicts between the underlying mathematical background and the capabilities expected by various application areas. The incorporation and integration of results from other mathematical disciplines than numerical analysis is expected, not only to improve the fundamentals, but also to lead to new and more intuitive design tools and methodologies.

The most common representation for curves and surfaces in CAD is the rational parametric representation, and often, the implicit algebraic representation for some specific operations. However, only the implicit algebraic curves and surfaces are closed under some basic geometric operations (such as offsetting or bisecting), while rational parametric representations are not conserved. The computation of their exact algebraic implicit representation is very complicated or impractical, since it often involves the representation and manipulation of the solution set of a system of nonlinear equations of high degree. The exact description for certain geometric

The authors are partially supported by the Spanish "Ministerio de Economia y Competitividad" and by the European Regional Development Fund (ERDF), under the Project MTM2011-25816-C02-02, and by the SAGA network.

I. Adamou • M. Fioravanti (✉) • L. Gonzalez-Vega
Departamento de Matematicas, Estadística y Computación, Universidad de Cantabria, Santander, Spain
e-mail: ibrahim.adamou@unican.es; mario.fioravanti@unican.es; laureano.gonzalez@unican.es

B. Mourrain
GALAAD, INRIA Méditerranée, Sophia-Antipolis, France
e-mail: Bernard.Mourrain@inria.fr

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_13

primitives, like offsets and bisectors, is very scarce, and therefore their use in CAD is still based on approximate methods, which might not be accurate enough. In this work, we present new methods for solving the following three problems:

1. Compute an exact parametrization of the bisector of two rational plane curves.
2. Compute exact parameterizations of the bisector of two rational surfaces.
3. Compute the topology of the Voronoï diagram of a finite family of parallel half-lines.

Bisector of Two Planar Curves

Consider two geometric objects O_1 and O_2 (points, parametric curves, parametric surfaces) in the space $\mathbb{R}^d, d \in \{2, 3\}$, with respective parameterizations $\mathbf{O}_1(u), u \in I_1$ and $\mathbf{O}_2(v), v \in I_2$, where I_1 and I_2 are intervals or rectangles (eventually reduced to a point).

Their (true) bisector $\mathcal{B}(O_1, O_2)$ is defined as the equidistant set of points from the two objects, i.e.:

$$\mathcal{B}(O_1, O_2) = \left\{ B \in \mathbb{R}^d : \inf_{u \in I_1} \|B - \mathbf{O_1}(u)\| = \inf_{v \in I_2} \|B - \mathbf{O_2}(v)\| \right\} . \tag{13.1}$$

The computation of an algebraic representation for the bisector is not an obvious task from the definition (13.1). Indeed, even if the parametric object is given with a regular and proper parametrization, it should be noted that the distance function

$$d : (B, O) \mapsto \inf_{u \in I} \|B - \mathbf{O}(u)\|, \tag{13.2}$$

is not always differentiable with respect to the point B, and a minimum of the distance function could be achieved at more than one parameter value. To overcome these difficulties, the following notion of untrimmed bisector is introduced (see [24, 36, 51]).

Definition 13.1 The untrimmed bisector of O_1 and O_2 is defined as the set of centres of spheres/circles which are tangent to O_1 and O_2 simultaneously.

This definition does not imply the same minimum distances measured from the two objects, in the presence of critical shapes on the objects (singular point, inflection point, self-intersection point). There are some extraneous parts that should be trimmed in order to achieve the true bisector. The trimming process (identifying and eliminating extraneous parts), which is the second challenge of the topic, is more convenient and effective from the untrimmed bisector parametrization when it is available.

There are several pairs of objects possessing a rational bisector, but in general it is very difficult to have a criterion for the rationality of the bisector, and very few generic configurations of objects with rational bisector are known. The planar curve-point bisector is rational, and an algorithmic method for the trimming is introduced by Farouki and Johnstone in [24]. However, the bisector of two coplanar curves is not rational in general (see [25]). On the other hand, it is expected the rationality of the bisector of two curves with a parametrization for which the norm of the speed vector is rational, called PH-curves (for Pythagorean Hodograph-curves) (see [53] and [45]). The bisector of a circle or a straight line and a PH-curve is rational (see [51]). The curve-curve bisector is an algebraic curve which, very often, is of very high degree (see [15]) so that the trimming becomes highly problematic, and impracticable. Thus, many ways of approximation have been proposed by researchers for the representation of the bisector curve (see [26] and [15]). Related to the bisector of two plane curves, we present in this paper the following methods:

1. A new algorithmic approach for computing the algebraic parametrization (rational or not rational) for the untrimmed bisector of two planar curves by using Cramer's rule and an elimination step. We analyze in detail the application of the algorithm to determine a parametrization of the bisector of two curves when one of them is a circle or a straight line.
2. A new automatic approach for geometric and numerical characterization of planar point-curve and curve-curve true bisector by using dynamic color in GeoGebra,[1] a dynamic mathematics software. The approach consists in scanning and displaying, in a specific chosen color, the geometric locus of the true bisector. Like the algorithmic method introduced by Shou et al. [59] for computing the bisector of a point and a plane algebraic curve, this approach does not need a trimming process. Furthermore, this approach allows to collect the coordinates of many points on the true bisector, up to a tiny error.

Bisector of Two Surfaces

In the case of two rational surfaces, most of the known methods for computing the exact description of the bisector are devised only for those bisectors possessing rational parametrizations. Various approaches are appropriately used for very special cases to determine a rational representation for the bisector (see for example [13]). In other cases, symmetry considerations reduce the bisector computation to the following cases: point-line, point-surface, or curve-surface [16, 17], where the bisector is a rational surface. Similarly to the case of curves, if the considered surfaces are PN-surfaces (for Pythagorean Normal-surfaces), i.e., surfaces with a parametrization such that the norm of the normal vector is rational (see [53] and [46]), the bisector is expected to be rational. Using Laguerre geometry, Martin

[1] http://geogebra.org

Peternell [51, 52] has shown the rationality of the bisector between: plane and PN-surfaces, two PN-developable surfaces, two canal surfaces and some other cases.

In the case of non-rationality of the bisector, the exact description is scarce, the implicit representation is of very high degree, and often, it is impractical to compute it. Thus, B-spline and other kinds of approximation have been proposed (see for instance [18]).

Our contribution for this case is a new approach, generalizing the method used in the two planar curves case. It uses the so-called generalized Cramer's rule (see [12] and [11]) and suitable elimination steps, for computing an algebraic parametrization (rational or non-rational) for the bisector surface of two low degree rational surfaces. Some of the obtained results coincide with those mentioned above in the rational case. The method is well-suited for approximation purposes, which is of special interest in the non-rational case.

The new introduced approach allows to easily obtain parametrizations of the plane-quadric, plane-torus, circular cylinder-quadric, circular cylinder-torus, cylinder-cylinder, cylinder-cone and cone-cone bisectors, which are rational in most cases. In the remaining cases, the parametrization involves square roots.

For the case of plane and quadric, or plane and torus, the implicit equation of the (untrimmed) bisector can be easily computed, either from the equations defining the bisector, or from the computed parametrization. The results we obtained for the parametrization, as well as for the implicit equation, assuming that the quadric is provided with its PN-parameterization (if possible), are summarized in Table 13.1. They coincide with those proved, using a different approach, by Peternell [51]. Note that the degree of the implicit equation of the untrimmed bisector is usually bigger than the degree of the true bisector.

Table 13.1 The algebraic representation of plane-quadric and plane-torus untrimmed bisectors

Plane-quadric/torus	Parametrization	Max. degree of implicit eqn.
Parabolic cylinder	Rational	6
Circular cylinder	Rational	4
Elliptic cylinder	Non rational	8
Hyperbolic cylinder	Non rational	8
Circular cone	Rational	4
Elliptic cone	Non rational	8
Sphere	Rational	4
Ellipsoid	Rational	12
Elliptic paraboloid	Rational	10
Hyperbolic paraboloid	Rational	10
Hyperboloid one-sheet	Rational	12
Hyperboloid two-sheets	Rational	12
Torus	Rational	8

Voronoï Diagram of a Family of Parallel Half-Lines

The Voronoï diagram (VD) is a fundamental data structure in computational geometry with many and very diverse applications in theoretical and practical areas, such as proximity queries, high-clearance placements, motion planning problems, classification problems, and many more (see [4, 27, 50] and [38]). For a given discrete set of geometric objects (called `sites`) $\mathcal{S} = \{s_1, \ldots, s_n\}$ in a metric space $(\mathcal{E}, \mathrm{d})$, the VD subdivides $\mathcal{E}$ in regions (called `cells`), where each region associated to one site s_i consists of the points closer to the site s_i than to any other site in $\mathcal{S}$, for the distance d. Generally, the VD can be defined as the Minimization Diagram (MD) (a projection of the lower envelope) of a finite set of continuous functions $\{f_1, \ldots, f_n\}$, where each function f_i is defined as the distance function to a site s_i (see [8]). Different types of sites and distance functions have been proposed for different kinds of applications, according to the context.

In the space $\mathbb{R}^2$ the VD has been studied extensively, and its structure is nowadays thoroughly well understood. Many robust and efficient algorithms have been proposed by several authors such as [19, 20, 23, 34, 56]. However, in three and higher dimension spaces, the VD has been much less studied, and many basic problems are still wide open. Some recent works for linear, quadrics, and semi-algebraic sites have been developed (see [4, 7, 31, 37, 38, 40, 42, 50] and [22]).

In this paper, we consider the VD of a set of n parallel half-lines, $\{d_1, \ldots, d_n\}$, oriented to the positive $x-$direction, i.e.:

$$d_i = (x_i + t, y_i, z_i), t \geqslant 0,\ i = 1 \ldots n, \tag{13.3}$$

where $x_l \neq x_k$, and $(y_l, z_l) \neq (y_k, z_k)$, $\forall l \neq k$, constrained to a compact domain $\mathcal{D}_0 \subset \mathbb{R}^3$ of the form:

$$\mathcal{D}_0 = [a_{1_0}, b_{1_0}] \times [a_{2_0}, b_{2_0}] \times [a_{3_0}, b_{3_0}]\,, \tag{13.4}$$

with respect to the Euclidean distance.

The VD of this new kind of sites, could be used to provide useful solutions to some theoretical and practical problems in the drilling industry (mining, offshore drilling, hydraulic, etc.), in particular for exploring drilling-deepwater closest ground, well collision and identifying unplanned ground avoidance (see [5, 29, 32, 60–62] and [41]).

Two general classes of algorithms are available for computing VD: exact and approximate. Some of the exact algorithms require the manipulation of an exact representation of VD boundaries, dealing with high-degree curves and surfaces, and their intersections, which lead very often to complex and very hard to implement algorithms (see [57]). In order to overcome these difficulties, approximate algorithms have been used often (see, for instance, [7]). We present a new approximate algorithm to determine a meshing for the VD of the kind of sites we consider. The method has three main parts. First, the subdivision of the initial domain $\mathcal{D}_0$ into sub-domains (boxes) until the topology of the boundary of the VD in each sub-domain is fully determined from its intersection with the boundary of the box, or the size of the sub-domain is smaller than a constant threshold ε initially fixed. Second, the meshing of the cell boundaries contained in each sub-domain. Finally, the reconstruction of the VD cells. Thus, the correct topology of the result is certified, with the possible exception of certain boxes of size smaller than ε. The representation of the output from the method for a family of four half-lines is represented in the Figs. 13.11–13.15. The implementation of the method has not been completed yet. The testing of the implemented subroutines produce accurate results with good timing.

The rest of the paper is organized in three sections, where we explain the details of the methods, state some general results, and give examples of the application of the algorithms. Section 13.2 is on the parametrization of the bisector of two curves, and the bisector of two surfaces. In Sect. 13.3 we describe the algorithm for computing the Voronoï diagram of a family of parallel half-lines, and the last section is devoted to conclusions.

13.2 Bisectors

This section is divided in three parts. First, we present the equations that characterize the untrimmed bisector of two curves in the plane, and the method to solve them. Second, we present the method for the graphical visualization and numerical data computation of the true bisector of two planar curves, using GeoGebra. In the last subsection, we present the algorithm for the parametrization of the untrimmed bisector of two surfaces.

13.2.1 Parametrization of Curve-Curve and Point-Curve Bisectors

Consider two plane rational curves (or a point and a curve). Starting from the equations of the untrimmed bisector of the curves (see Definition 13.1), we will present a method for computing its parametrization. We will analyze the special cases in which one of the curves is either a straight line or a circle.

Equations of the Untrimmed Bisector Curve

Let s and r be two regular rational curves, and $\mathbf{s}(u)$ and $\mathbf{r}(t)$ their parametrizations, respectively. A point $\mathbf{B} = (X, Y) \in \mathbb{R}^2$ is in the untrimmed bisector of the curves s and r if it satisfies the following system of equations (see [36]):

- The point $\mathbf{B}$ is in the normal lines of s and r, at $\mathbf{s}(u)$ and $\mathbf{r}(t)$, respectively:

$$\begin{aligned} \langle (X,Y) - \mathbf{s}(u), \mathbf{s}'(u) \rangle &= 0, \\ \langle (X,Y) - \mathbf{r}(t), \mathbf{r}'(t) \rangle &= 0, \end{aligned} \tag{13.5}$$

  where $\mathbf{s}'$ and $\mathbf{r}'$ denote derivatives.
- The point $\mathbf{B}$ is at equal distance from $\mathbf{s}(u)$ and $\mathbf{r}(t)$:

$$\langle (X,Y), 2\,(\mathbf{r}(t) - \mathbf{s}(u)) \rangle + \|\mathbf{s}(u)\|^2 - \|\mathbf{r}(t)\|^2 = 0. \tag{13.6}$$

Equations (13.5) can be written in matrix form as follows

$$\mathbf{A}\,\mathbf{B}^T = \mathbf{V}, \tag{13.7}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{s}'_x(u) & \mathbf{s}'_y(u) \\ \mathbf{r}'_x(t) & \mathbf{r}'_y(t) \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} \langle \mathbf{s}(u), \mathbf{s}'(u) \rangle \\ \langle \mathbf{r}(t), \mathbf{r}'(t) \rangle \end{bmatrix}.$$

The Algorithm for Computing a Parametrization of the Untrimmed Bisector

Our goal is to compute a parametrization of the untrimmed bisector of the curves s and r in terms of one parameter, either u or t. Our approach consists in:

- First, solve the system (13.7) for $\mathbf{B}$ in terms of u, t, using Cramer's rule, and substitute $\mathbf{B}(u,t)$ in (13.6), to obtain the equation:

$$F(u,t) = \langle \mathbf{B}(u,t), 2(\mathbf{r}(t) - \mathbf{s}(u)) \rangle + \|\mathbf{s}(u)\|^2 - \|\mathbf{r}(t)\|^2 = 0. \tag{13.8}$$

- Then, if possible, express one of the parameters, say u in terms of t, from Eq. (13.8). More precisely, if $F(u,t)$ is a product of linear expressions in u, one can easily express u in terms of t. Since there might be more than one solution according to the degree of $F(u,t)$ with respect to u, we get

$$u_i = u_i(t), \ i = 1, \dots, m.$$

- Finally, substitute u by $u_i(t)$ in $\mathbf{B}(u,t)$, for each solution, and obtain the parametrization of the untrimmed bisector of the form:

$$\mathbf{b}_i(t) = \mathbf{B}(u_i(t), t) = [x_i(t), y_i(t)]^T,$$

  where $x_i(t), y_i(t)$ are in general non-rational.

Remark 13.1 Although the parametrization we obtain is not rational in general, we may apply the trimming method of Farouki and Johnstone [24], which is based in finding critical points (self-intersections, cusps, etc.) of the untrimmed bisector, and identifying the intervals that must be trimmed. From the parametrization of the bisector obtained with our method, we can compute the critical points for this purpose.

We apply our algorithm, to parametrize the bisector of two rational curves, in the particular cases where one of the curves is either a line or a circle.

1. **Line-curve bisector**: Assume s is a line and r is a regular rational curve, with the respective parametrizations

$$\mathbf{s}(u) = (\alpha_1\, u + \beta_1\,,\, \alpha_2\, u + \beta_2) \quad \text{and} \quad \mathbf{r}(t) = (a(t)\,,\, b(t))\,. \tag{13.9}$$

Applying the algorithm to this particular case we get the following result. The theorem follows from straightforward computation. Moreover, we obtain a closed form expression of the parametrizations in terms of $\alpha_1, \beta_1, \alpha_2, \beta_2, a(t), b(t), a'(t)$, and $b'(t)$. For full details see [1], Section 2.2.3.

Theorem 13.1 *Let s and r be as in (13.9), and let $\sigma(t) = \sqrt{a'(t)^2 + b'(t)^2}$ be the norm of the speed vector of the curve r.*

(a) The equation $F(u,t) = 0$ is quadratic in u, and its discriminant with respect to u is non-negative. Therefore there exist two components $\mathbf{b_1}$ and $\mathbf{b_2}$ for the untrimmed bisector parametrization, which might contain a square root of a non-negative polynomial expression (originating from σ).

(b) If the curve r is a PH-curve, the untrimmed bisector parametrization is rational. Furthermore, if the norm of the speed vector of the line s,

$$\|\mathbf{s}'(u)\| = \sqrt{\alpha_1^2 + \alpha_2^2}$$

is not a rational number, then the parametrizations $\mathbf{b_1}, \mathbf{b_2}$ of the untrimmed bisector are rational expressions with coefficients in the field $\mathbb{Q}\left[\sqrt{\alpha_1^2 + \alpha_2^2}\right]$.

2. **Circle-curve bisector:** Assume s is a circle of radius d, and r is a regular rational curve, with the respective parametrizations

$$\mathbf{s}(u) = \left(\frac{2\,d\,u}{1+u^2}\,,\, \frac{d\,(1-u^2)}{1+u^2}\right) \quad \text{and} \quad \mathbf{r}(t) = (a(t)\,,\, b(t))\,. \tag{13.10}$$

If we apply the method to this case, Eq. 13.8 takes the form

$$F(u,t) = \left(1+u^2\right)^2 F_0(u,t) = 0,$$

where

$$\begin{aligned}F_0(u,t) = &-d^2u^2b'(t) + a(t)^2b'(t)u^2 - 2\,a'(t)a(t)du^2 - 2\,b'(t)b(t)d\,u^2\\&-b'(t)b(t)^2u^2 - 2\,a'(t)a(t)b(t)u^2 - 2\,b(t)^2ua'(t)\\&+2\,ua'(t)a(t)^2 + 2\,d^2ua'(t) + 4\,ub'(t)b(t)a(t) + d^2b'(t)\\&-2\,b'(t)b(t)d + 2\,a'(t)a(t)b(t) - 2\,a'(t)a(t)d\\&-a(t)^2b'(t) + b'(t)b(t)^2,\end{aligned}$$

is quadratic in u. Therefore, we get the following result.

Theorem 13.2 *Let s be a circle and r a rational plane curve, with the parametrizations (13.10). Let $\sigma(t) = \sqrt{a'(t)^2 + b'(t)^2}$, be the norm of the speed vector of r.*

(a) The discriminant of the equation $F(u,t) = 0$, with respect to u is non-negative. Therefore, there exist two components $\mathbf{b_1}$ and $\mathbf{b_2}$ of the untrimmed bisector parametrization, which might contain a square root of a non-negative polynomial expression (originating from σ).

(b) If the curve r is a PH-curve, the untrimmed bisector parametrization is rational.

Proof The theorem follows directly by completing the computations, and a closed form expression of the parametrizations is obtained. See [1], Section 2.2.4, for full details.

Remark 13.2 The definition of untrimmed bisector is invariant under Möbius transformations. If we have a bisector parametrization $\mathbf{B}$ of two objects O_1, O_2, and a Möbius transformation M, then $M(\mathbf{B})$ will be a parametrization of the bisector of $M(O_1)$ and $M(O_2)$. This implies, in particular, that Theorem 13.2 follows from Theorem 13.1, because given a circle and a straight line, there is a Möbius transformation that maps the circle onto the line. This provides an alternative method for producing certain parametrizations.

A similar reasoning can be applied in tridimensional space to relate, using Möbius transformations, sphere-surface bisectors with plane-surface bisectors.

Example 13.2.1 *Let*

$$\mathbf{s}(u) = \left(\frac{2\,u}{1+u^2}, \frac{1-u^2}{1+u^2} + 4\right) \quad \textit{and} \quad \mathbf{r}(t) = \left(2\,t^2 - 2,\, \frac{2}{3\sqrt{3}}\,t\left(t^2-1\right)\right),$$

be the parametrizations of a circle and a PH-cubic, respectively. The norms of the respective speed vectors of s and r are given by:

$$\sigma_{\mathbf{s}}(u) = \frac{2}{u^2+1} \quad \textit{and} \quad \sigma_{\mathbf{r}}(t) = \frac{2\sqrt{3}}{3}\left(1 + 3\,t^2\right).$$

Applying the algorithm to $\mathbf{s}(u)$ *and* $\mathbf{r}(t)$ *we have:*

$$\mathbf{B}(u,t) = \begin{bmatrix} \dfrac{4\,u\left(-2\,\sqrt{3}+6\,\sqrt{3}t^2+5\,t-2\,t^3-3\,t^5\right)}{\sqrt{3}-3\,\sqrt{3}t^2-\sqrt{3}u^2+3\,\sqrt{3}t^2u^2-12\,tu} \\ \dfrac{2\,t\left(-24\,u+5-2\,t^2-3\,t^4-5\,u^2+2\,u^2t^2+3\,u^2t^4\right)}{\sqrt{3}-3\,\sqrt{3}t^2-\sqrt{3}u^2+3\,\sqrt{3}t^2u^2-12\,tu} \end{bmatrix},$$

$$F(u,t) = -\frac{\sqrt{3}}{3}\left(u^2+1\right)^2 F_0(u,t) = 0,$$

where

$$\begin{aligned} F_0(u,t) = {} & \left(6t^4u+4\sqrt{3}t^3u-13\sqrt{3}tu+3u+6t^4-4\sqrt{3}t^3-11\sqrt{3}t-21\right) \\ & \left(-6t^4u+4\sqrt{3}t^3u+5\sqrt{3}tu+15u+6t^4+4\sqrt{3}t^3-19\sqrt{3}t+9\right). \end{aligned}$$

The two expressions of u in terms of t are given by:

$$\begin{aligned} u_1(t) &= -\frac{6\,t^4+4\,\sqrt{3}t^3-19\,\sqrt{3}t+9}{-6\,t^4+4\,\sqrt{3}t^3+5\,\sqrt{3}t+15}, \\ u_2(t) &= \frac{-6\,t^4+4\,\sqrt{3}t^3+11\,\sqrt{3}t+21}{6\,t^4+4\,\sqrt{3}t^3-13\,\sqrt{3}t+3}, \end{aligned}$$

and the untrimmed bisector parametrization is given by:

$$\mathbf{b_1}(t) = \mathbf{B}(u_1(t),t), \quad \mathbf{b_2}(t) = \mathbf{B}(u_2(t),t).$$

The true bisector has two branches, one is given by the restriction of the component $\mathbf{b_1}$ *to the interval* $[-0.8331, -0.0697]$ *and the other by the restriction of the component* $\mathbf{b_2}$ *to the interval* $[1.16838, 3.1916]$.

The two curves, the untrimmed and the trimmed bisectors are shown in the Figs. 13.1 and 13.2.

13.2.2 Geometric and Numerical Characterization for the Bisector using GeoGebra Dynamic Color

GeoGebra is a free software program (http://www.geogebra.org), that combines Dynamic Geometry with the algebraic equations of the geometric objects (points, lines, conics, etc.) represented. It gives simultaneously three representations of the same mathematical object: geometric (graphical), algebraic and as cell of a spreadsheet; any change in one of the representations produces, in a dynamic

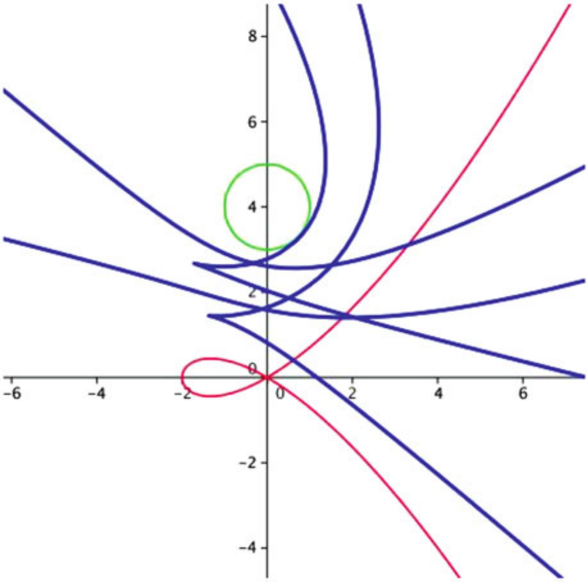


Fig. 13.1 The curves (*green* and *red*) and the untrimmed bisector (*blue*) of Example 13.2.1

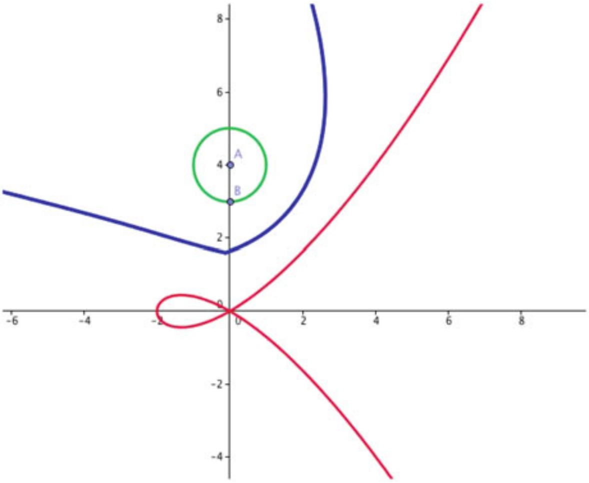


Fig. 13.2 The curves (*green* and *red*) and the trimmed bisector (*blue*) of Example 13.2.1

way, the corresponding change in the other representations. Using color models [30,49], the GeoGebra automatic scanning method for an unknown geometric locus is applied for solving the problem of the true bisector of two planar geometric objects (curve, point) O_1 and O_2. Furthermore, the method allows the numerical collection of coordinates of a sample of true bisector points, up to a very small fixed error ε. The work of this subsection has been done by the first author in collaboration with R. Losada.

A color model (for example, RGB) is a specification of a three dimensional coordinate system where each color is represented by a single point. The scanner consists in a sweeping vertical line, with a dense sample of n points on each vertical line. The scanner line is automatically animated at the touch of a button, proceeding in a horizontal pixel displacement and sweeping a domain containing the two geometric objects. As a result of the process, each sampled point in the screen will acquire a different color, depending on its coordinates. A good example of application of color to the study of a geometric problem is presented in [44].

1. **Geometric characterization of the true bisector curve**

   In order to use the scanning process, it is sufficient to know the condition that each point B of the bisector locus must obey, i.e:

$$\mathrm{D}(B) \equiv \mathrm{d}(B, O_1) - \mathrm{d}(B, O_2). \tag{13.11}$$

   For the distances from each scanner point to O_1 and O_2 the GeoGebra command used is:

$$\text{Distance}\,[< \textit{point} >, < \textit{object} >]\,,$$

   which computes the minimal distance from a point to an object, as required in the bisector definition (13.1).

   Using the **RGB** model, the three dynamic color fields (Red, Green, Blue) associated to each point B will be given the value

$$\exp(-|\mathrm{D}(B)|),$$

   which normalizes any value of (13.11) to the interval $(0, 1]$, such that, if $\mathrm{D}(B)$ is 0, the resulting **RGB** value will be $(1, 1, 1)$ (i.e., white color), wherein the closer of the pure white for the lower absolute value of $\mathrm{D}(B)$. Thus, the true bisector curve is geometrically characterized in white color.

   Example 13.2.2 *(a) Let*

$$\mathbf{C}_1(t) = \left(t^2, t^3\right) \quad \textit{and} \quad \mathbf{C}_2(t) = \left(t^2 + 5\,,\, t\right),$$

   be the respective parametrizations of a cubic and a parabola curves. For $n = 290$ *the geometric characterization of their true bisector curve is presented in Fig. 13.3.*

   (b) Let

$$\mathbf{C_1}(u) = \left(u^3 + 5\,,\, u^2\right) \quad \textit{and} \quad P = \left(10\,,\, -1\right),$$

   be respectively the parametrization of a cubic curve and a fixed point. For $n = 290$ *the geometric characterization of their true bisector curve is presented in Fig. 13.4.*

2. **Numerical characterization of the true bisector curve**

   Let O_1 and O_2 be two planar geometric objects, and let ε be a small enough fixed constant. During the scanning process of the true bisector locus of O_1 and

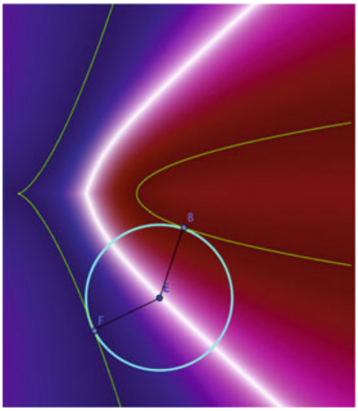


Fig. 13.3 The geometric characterization of the true bisector from Example 13.2.2(a) in *white color*

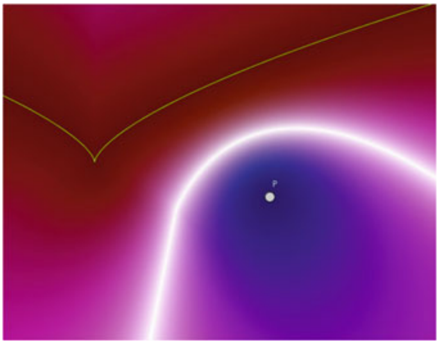


Fig. 13.4 The geometric characterization of the true bisector from Example 13.2.2(b) in *white color*

O_2, the coordinates of each scanned point B_i verifying

$$\left|\mathrm{d}(B_i, O_1) - \mathrm{d}(B_i, O_2)\right| < \varepsilon, \tag{13.12}$$

are stored in a list and their traces are plotted. When the error is very little, we obtain a small list of the coordinates of the scanned points (see the example below).

Example 13.2.3 *Let*

$$\mathbf{C}_1(t) = \left(\frac{1}{4}t^2 - 5\,,\, t\right) \quad \textit{and} \quad \mathbf{C}_2(t) = \left(\frac{2(1-t^2)}{1+t^2}\,,\, \frac{2t}{1+t^2}\right),$$

be respectively the parametrizations of a parabola and an ellipse. As a result of the scanning process with GeoGebra, we get the following:

(a) For $n = 400$ and $\varepsilon = 0.005$, we obtain a list of the coordinates of 134 points, whose traces are plotted in Fig. 13.5,

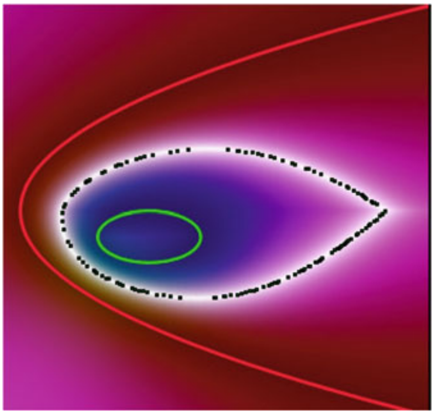


Fig. 13.5 The trace of the true bisector points in *black* for $n = 400$ and $\varepsilon = 0.005$

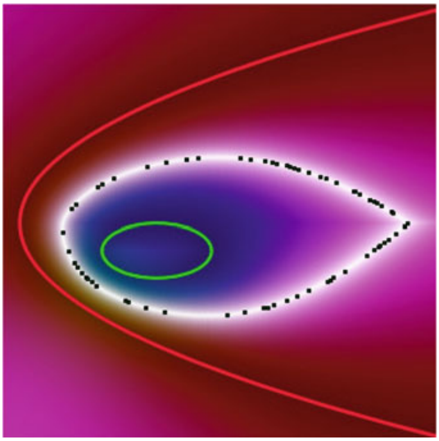


Fig. 13.6 The trace of the true bisector points in *black* for $n = 400$ and $\varepsilon = 0.0025$

(b) *For* $n = 400$ *and* $\varepsilon = 0.0025$*, we obtain a list of the coordinates of 71 points, whose traces are plotted in Fig. 13.6,*

(c) *For* $n = 400$ *and* $\varepsilon = 0.0005$ *we obtain a list of the coordinates of 38 points, whose traces are plotted in Fig. 13.7.*

13.2.3 The Bisector Surface of Two Low Degree Rational Surfaces

Generalizing the method of Sect. 13.2.1 we present an algorithm to compute the untrimmed bisector surface of two rational surfaces (see Definition 13.1). We start by giving the equations which characterize the bisector.

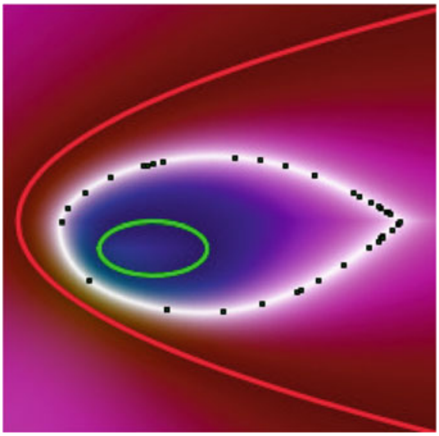


Fig. 13.7 The trace of the true bisector points in *black* for $n = 400$ and $\varepsilon = 0.0005$

Equations of the Untrimmed Bisector Surface

Let S_1 and S_2 be two regular rational surfaces, and $\mathbf{S}_1(s,t)$ and $\mathbf{S}_2(u,v)$ their parametrizations, respectively. A point $\mathbf{B} = \big(X, Y, Z\big) \in \mathbb{R}^3$ is in the bisector of the surfaces S_1 and S_2 if it satisfies the following system of equations (see [36]):

- The point $\mathbf{B}$ is in the normal lines of S_1 and S_2, at $\mathbf{S}_1(s,t)$ and $\mathbf{S}_2(u,v)$, respectively:

$$\begin{aligned} \langle (X,Y,Z) - \mathbf{S}_1(s,t), \partial_s \mathbf{S}_1(s,t)\rangle &= 0,\\ \langle (X,Y,Z) - \mathbf{S}_1(s,t), \partial_t \mathbf{S}_1(s,t)\rangle &= 0,\\ \langle (X,Y,Z) - \mathbf{S}_2(u,v), \partial_u \mathbf{S}_2(u,v)\rangle &= 0,\\ \langle (X,Y,Z) - \mathbf{S}_2(u,v), \partial_v \mathbf{S}_2(u,v)\rangle &= 0, \end{aligned} \tag{13.13}$$

- The point $\mathbf{B}$ is at equal distance from $\mathbf{S}_1(s,t)$ and $\mathbf{S}_2(u,v)$:

$$\langle (X,Y,Z), 2(\mathbf{S}_2(u,v) - \mathbf{S}_1(s,t))\rangle + \|\mathbf{S}_1(s,t)\|^2 - \|\mathbf{S}_2(u,v)\|^2 = 0. \tag{13.14}$$

Equations (13.13) can be written in matrix form as follows

$$\mathbf{A}\,\mathbf{B}^T = \mathbf{R}, \tag{13.15}$$

where

$$\mathbf{A} = \begin{bmatrix} \partial_s \mathbf{S}_{1x} & \partial_s \mathbf{S}_{1y} & \partial_s \mathbf{S}_{1z} \\ \partial_t \mathbf{S}_{1x} & \partial_t \mathbf{S}_{1y} & \partial_t \mathbf{S}_{1z} \\ \partial_u \mathbf{S}_{2x} & \partial_u \mathbf{S}_{2y} & \partial_u \mathbf{S}_{2z} \\ \partial_v \mathbf{S}_{2x} & \partial_v \mathbf{S}_{2y} & \partial_v \mathbf{S}_{2z} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \langle \mathbf{S}_1, \partial_s \mathbf{S}_1\rangle \\ \langle \mathbf{S}_1, \partial_t \mathbf{S}_1\rangle \\ \langle \mathbf{S}_2, \partial_u \mathbf{S}_2\rangle \\ \langle \mathbf{S}_2, \partial_v \mathbf{S}_2\rangle \end{bmatrix}.$$

The Algorithm for Computing the Bisector Parametrization

Our goal is to compute a parametrization of the bisector of S_1 and S_2 in terms of two parameters chosen from $u,\ v,\ s$ and t. If for some values of the parameters $u,\ v,\ s,\ t$, the linear system (13.15) of unknown $\mathbf{B}$ admits a solution, then $\operatorname{rank}(|\mathbf{A},\mathbf{R}|) \leqslant 3$, and thus $\det(|\mathbf{A},\mathbf{R}|) = 0$.

Our approach consists in:

- First solve the system (13.15) for $\mathbf{B}$ in terms of u, v, s, t, by using the Moore-Penrose generalized inverse of $\mathbf{A}$, $\mathbf{A}^\dagger$ (see [11]):

$$\mathbf{A}^\dagger = -a_k^{-1}\mathbf{A}^T\left[(\mathbf{A}\mathbf{A}^T)^{k-1} + a_1(\mathbf{A}\mathbf{A}^T)^{k-2} + \ldots + a_{k-1}I_n\right],$$

where a_i and k are such that

$$\det(wI_n - \mathbf{A}\mathbf{A}^T) = a_0w^n + a_1w^{n-1} + \ldots + a_{n-1}w + a_n,$$

with $a_0 = 1$; ($k \neq 0$ is the largest index such that $a_k \neq 0$) or ($k = 0$ and $\mathbf{A}^\dagger = \mathbf{0}$). Thus

$$\mathbf{B}^T(u,v,s,t) = \mathbf{A}^\dagger\mathbf{R}\,.$$

Substituting $\mathbf{B}(u,v,s,t)$ in (13.14) we obtain the equation:

$$F(u,v,s,t) = \langle\mathbf{B}, 2(\mathbf{S}_2(u,v) - \mathbf{S}_1(s,t))\rangle + \|\mathbf{S}_1(s,t)\|^2 - \|\mathbf{S}_2(u,v)\|^2. \qquad (13.16)$$

- Then, eliminate two of the four parameters u, v, s and t from

$$G(u,v,s,t) = \det(|\mathbf{A},\mathbf{R}|) = 0, \quad F(u,v,s,t) = 0, \qquad (13.17)$$

as follows:

  – First, if possible, express one of the parameters, say t, in terms of u, v and s from

$$G(u,v,s,t) = 0.$$

More precisely, if $G(u,v,s,t)$ is a polynomial linear (or a factor of k linear expressions) in t, one can easily express t in terms of u, v and s, namely $t = t(u,v,s)$ (or $t_j = t(u,v,s)$, $j = 1,\ldots,k$), and substitute in

$$F(u,v,s,t) = 0$$

to obtain

$$F_0(u,v,s) = F(u,v,t(u,v,s)) = 0,$$

- Then, in the same way, if possible, express s in terms of u and v from

$$F_0(u, v, s) = 0.$$

There might be more than one solution for s according to the degree of $F_0(u, v, s)$ in s, we get

$$s_i = s_i(u, v), i = 1, \dots, m;$$

- Finally, substitute successively t by $t(u, v, s)$, and for each solution, s by $s_i(u, v)$, in $\mathbf{B}(u, v, s, t)$. We obtain the untrimmed bisector parametrization of the form:

$$\mathbf{B}_i(u, v) = \mathbf{B}^T\ (u, v, s_i(u, v), t(u, v, s_i(u, v))) = \begin{bmatrix} x_i(u, v) \\ y_i(u, v) \\ z_i(u, v) \end{bmatrix},$$

where $x_i(u, v), y_i(u, v), z_i(u, v)$ are in general non-rational.

Using this algorithm, we can deal with the specific cases of plane-quadric, plane-torus, circular cylinder-non developable quadric, circular cylinder-torus, cylinder-cylinder, cylinder-cone and cone-cone bisectors.

Remark 13.3 In general, the rationality of the bisector parameterizations $\mathbf{B}_i(u, v)$ depend on the rationality of the norm of the normal vectors of the two surfaces, $\sigma_1(s, t) = \|\partial_s S_1 \times \partial_t S_1\|$ and $\sigma_2(u, v) = \|\partial_u S_2 \times \partial_v S_2\|$. Since σ_1 and σ_2 could be rational functions with non-rational coefficients, i.e.

$$\exists\, \delta_1, \delta_2 \notin \mathbb{Q}, \sigma_1 \in \mathbb{Q}[\delta_1](s, t), \sigma_2 \in \mathbb{Q}[\delta_2](u, v),$$

then the bisector parameterizations $\mathbf{B}_i(u, v)$ could also be rational with non rational coefficients. More precisely if $\sigma_1 \in \mathbb{Q}[\delta_1](s, t)$ and $\sigma_2 \in \mathbb{Q}[\delta_2](u, v)$, then $\mathbf{B}_i \in \mathbb{Q}[\delta_1 \cdot \delta_2](u, v)$.

We apply the method to compute the bisector of a plane and a quadric, or a plane and a torus.

1. **Plane and quadric/torus**

   Let

$$\mathbf{P}(s, t) = \begin{bmatrix} \alpha_1 s + \beta_1 t + \gamma_1 \\ \alpha_2 s + \beta_2 t + \gamma_2 \\ \alpha_3 s + \beta_3 t + \gamma_3 \end{bmatrix} \quad \text{and} \quad \mathbf{S}(u, v) = \begin{bmatrix} S_x(u, v) \\ S_y(u, v) \\ S_z(u, v) \end{bmatrix}, \tag{13.18}$$

be the respective parametrizations of a plane and a quadric or a torus. Since $\mathbf{A}$ does not depend on s and t, and $\mathbf{R}$ is linear in s and t, then $G(u, v, s, t)$ is linear in s and t, and $F(u, v, s, t)$ is quadratic in s and t. Applying the algorithm, we have the following theorem.

Theorem 13.3 *Consider a plane and a quadric surface or a torus, with the parametrizations (13.18). Let $\sigma_p = \|\partial_s \mathbf{P} \times \partial_t \mathbf{P}\|$ and $\sigma_s(u, v) = \|\partial_u \mathbf{S} \times \partial_v \mathbf{S}\|$.*

(a) There exist two components of the parametrization for the untrimmed bisector $\mathbf{B}_i(u, v), i = 1, 2$, that might contain a square root of a non-negative expression (originating from σ_s).

(b) If $\sigma_s(u, v)$ is rational, then the untrimmed bisector parametrizations $\mathbf{B}_i(u, v), i = 1, 2$, are rational. Furthermore, if $\sigma_p \notin \mathbb{Q}$ and $\sigma_s \in \mathbb{Q}[\boldsymbol{\delta}](u, v)$, then $\mathbf{B_i} \in \mathbb{Q}[\boldsymbol{\sigma_p} \cdot \boldsymbol{\delta}](u, v)$ (see Example 13.2.4).

Proof From $G(u, v, s, t) = 0$ we get $s = s(u, v, t)$ and substitute in the equation $F(u, v, s, t) = 0$ to get

$$F_0(u, v, t) = F(u, v, s(u, v, t), t) = \omega_0 t^2 + \omega_1 t + \omega_2 = 0, \tag{13.19}$$

where ω_0, ω_1, ω_2 depend on α_i, β_i γ_i $\partial_u \mathbf{S}$, $\partial_v \mathbf{S}$, $\mathbf{S}$. After some computations, the discriminant of Eq. (13.19) is

$$\Delta = 4\left[(\langle \Gamma, (\partial_s \mathbf{P} \times \partial_t \mathbf{P})\rangle) \,.\, \langle(\partial_t \mathbf{P} \times (\partial_s \mathbf{P} \times \partial_t \mathbf{P})), (\partial_u \mathbf{S} \times \partial_v \mathbf{S})\rangle\right]^2 \\ \sigma_p^2 \sigma_s^2(u, v) \geqslant 0$$

where

$$\Gamma = \begin{bmatrix} S_x - \gamma_1 \\ S_y - \gamma_2 \\ S_z - \gamma_3 \end{bmatrix}, \ \sigma_p = \|\partial_s \mathbf{P} \times \partial_t \mathbf{P}\| \text{ and } \sigma_s(u, v) = \|\partial_u \mathbf{S} \times \partial_v \mathbf{S}\|.$$

Thus Eq. (13.19) has two solutions:

$$t_1(u, v) = \frac{-\omega_1 - \sqrt{\Delta}}{2\omega_0}, \quad t_2(u, v) = \frac{-\omega_1 + \sqrt{\Delta}}{2\omega_0},$$

with

$$\sqrt{\Delta} = 2\,|(\langle \Gamma, (\partial_s P \times \partial_t P)\rangle) \,.\, \langle(\partial_t P \times (\partial_s P \times \partial_t P)), (\partial_u S \times \partial_v S)\rangle|\,|\,\sigma_p \sigma_s(u, v)|.$$

Substitute first s by $s(u, v, t)$, and then t by $t_1(u, v)$ and $t_2(u, v)$, in $\mathbf{B}(u, v, s, t)$. Thus, the two components of the parametrization of the bisector are

$$\mathbf{B}_1(u, v) = \mathbf{B}(u, v, s(u, v, t_1), t_1) \text{ and } \mathbf{B}_2(u, v) = \mathbf{B}(u, v, s(u, v, t_2), t_2).$$

Since

$$|(\langle \Gamma, (\partial_s \mathbf{P} \times \partial_t \mathbf{P})\rangle)(\langle(\partial_t \mathbf{P} \times (\partial_s \mathbf{P} \times \partial_t \mathbf{P})), (\partial_u \mathbf{S} \times \partial_v \mathbf{S})\rangle)|$$

is a rational expression, and the norm of the normal vector $\sigma_p = \|\partial_s \mathbf{P} \times \partial_t \mathbf{P}\|$ is a constant, then the parametrization of the bisector is rational if $\sigma_s(u, v)$ is rational. In addition, if the constant $\sigma_p \notin \mathbb{Q}$ and if $\sigma_s(u, v) \in \mathbb{Q}[\delta](u, v)$, for a certain $\delta \notin \mathbb{Q}$, then the parametrization of the bisector is rational, i.e., $\mathbf{B}_i \in \mathbb{Q}[\sigma_p.\delta](u, v)$. □

Since all quadric surfaces are PN, except the non-circular cylinder or cone, we deduce the following corollary.

Corollary 13.1 *The bisector of a plane and a PN-surface is rational. In particular the bisectors of a plane and all quadric surfaces, except non-circular-cylinders and non-circular-cones, and the bisector of a plane and a torus, are rational.*

Example 13.2.4 *Consider a plane and a circular cone, respectively parametrized by*

$$P(s,t) = \begin{bmatrix} s \\ t \\ \frac{1}{3}(t+s) \end{bmatrix} \quad \textit{and} \quad S(u,v) = \begin{bmatrix} \dfrac{2\,uv}{1+u^2} \\ \dfrac{v\left(1-u^2\right)}{u^2+1} \\ v \end{bmatrix}.$$

Note that the cone axis is oblique to the plane. The norms of the normal vectors are

$$\sigma_p = \frac{\sqrt{11}}{3} \notin \mathbb{Q} \text{ and } \sigma_s = \frac{2\sqrt{2}v}{u^2+1} \in \mathbb{Q}[\sqrt{2}](u,v).$$

Applying the algorithm we obtain the two components of the untrimmed bisector parametrization:

$$\mathbf{B_1}(u,v) = \begin{bmatrix} -\dfrac{2uv\left(2\sqrt{22}u^2-5\,u^2-\sqrt{22}u+6\,u+\sqrt{22}+1\right)}{3+12\,u^2+9\,u^4-4\,u^3-8\,u} \\ \dfrac{v\left(2\sqrt{22}u^4-5u^4-\sqrt{22}u^3+6u^3-\sqrt{22}u^2+6u^2+\sqrt{22}u-6u-\sqrt{22}-1\right)}{3+12\,u^2+9\,u^4-4\,u^3-8\,u} \\ \dfrac{v\left(2\sqrt{22}u^4+13u^4-\sqrt{22}u^3-2u^3+3\sqrt{22}u^2+20u^2-\sqrt{22}u-10u+7+\sqrt{22}\right)}{3+12\,u^2+9\,u^4-4\,u^3-8\,u} \end{bmatrix},$$

$$\mathbf{B_2}(u,v) = \begin{bmatrix} \dfrac{2uv\left(2\sqrt{22}u^2+5\,u^2-\sqrt{22}u-6\,u+\sqrt{22}-1\right)}{3+12\,u^2+9\,u^4-4\,u^3-8\,u} \\ \dfrac{-v\left(2\sqrt{22}u^4+5u^4-\sqrt{22}u^3-6u^3-\sqrt{22}u^2-6u^2+\sqrt{22}u+6u-\sqrt{22}+1\right)}{3+12u^2+9u^4-4u^3-8u} \\ \dfrac{-v\left(2\sqrt{22}u^4-13u^4-\sqrt{22}u^3+2\,u^3+3\sqrt{22}u^2-20u^2-\sqrt{22}u+10u-7+\sqrt{22}\right)}{3+12u^2+9u^4-4u^3-8u} \end{bmatrix}.$$

The two surfaces and the bisector are shown in Fig. 13.8.

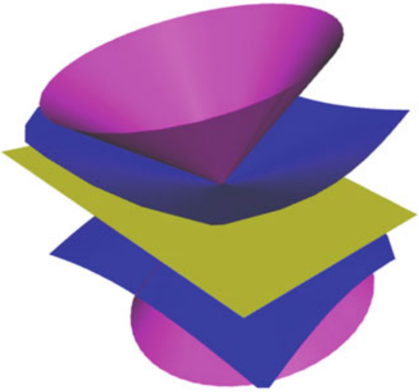


Fig. 13.8 Bisector of cone and oblique plane

2. **Circular cylinder and non developable quadric/torus**
   Let

$$\mathbf{C}(s,t) = \begin{bmatrix} \dfrac{2\,rs}{1+s^2} \\ \dfrac{r(1-s^2)}{1+s^2} \\ t \end{bmatrix} \quad \text{and} \quad \mathbf{S}(u,v) = \begin{bmatrix} S_x(u,v) \\ S_y(u,v) \\ S_z(u,v) \end{bmatrix},$$

be the respective parametrizations of a circular cylinder and a quadric or a torus. Note that in this case

$$G(u,v,s,t) = \det(|\mathbf{A},\mathbf{R}|) = 0$$

is linear in t and quadratic in s. More precisely

$$G(u,v,s,t) = \Theta_0(t)\,(s^2-1) + \Theta_1(t)\,s = 0, \tag{13.20}$$

where

$$\begin{aligned} \Theta_0(t) &= -2\,a(\partial_u S_y\,\partial_v S_z - \partial_u S_z\,\partial_v S_y)t - 2\,a(\partial_v S_y\,S_z\,\partial_u S_z - \partial_u S_y\,S_x\,\partial_v S_x \\ &\quad -\partial_u S_y\,S_z\,\partial_v S_z + \partial_v S_y\,S_x\,\partial_u S_x), \\ \Theta_1(t) &= -2\,a(-2\,\partial_u S_x\,\partial_v S_z + 2\,\partial_u S_z\,\partial_v S_x)t - 2\,a(2\,\partial_u S_x\,S_y\,\partial_v S_y \\ &\quad -2\,\partial_v S_x\,S_y\,\partial_u S_y + 2\,\partial_u S_x\,S_z\,\partial_v S_z - 2\,\partial_v S_x\,S_z\,\partial_u S_z). \end{aligned}$$

We distinguish two cases:

(a) The surface S is a surface of revolution, sharing the same axis with the cylinder C.

(b) The surface S is not of revolution or it does not share the same axis with the cylinder C.

The Surface S Is a Surface of Revolution, Sharing the Same Axis with the Cylinder C

If we parametrize the surface of revolution by

$$\mathbf{S}(u,v) = \begin{bmatrix} A(u)\frac{1-v^2}{1+v^2} \\ A(u)\frac{2v}{1+v^2} \\ B(u) \end{bmatrix}, \; A,\; B \in \mathbb{R}(u),$$

we get, using $G(u,v,s,t) = 0$,

$$s_1(v) = \frac{1-v}{1+v}, \quad s_2(v) = -\frac{1+v}{1-v}.$$

In both cases $F_i(u,v,t) = F(u,v,s_i(v),t)$ is quadratic in t. Therefore we obtain the following result. The complete proof can be found in [1], Section 3.4.

Theorem 13.4 *Consider a cylinder C and a surface of revolution S sharing the same axis with C. Let $\sigma_s(u,v) = \|\partial_u\mathbf{S} \times \partial_v\mathbf{S}\|$ be the norm of the normal vector of S.*

(a) There exist four components of the parameterization for the untrimmed bisector $\mathbf{B}_i(u,v), i = 1,2,3,4$, that might contain a square root of a non-negative expression (originating from σ_s).
(b) If σ_s is rational, then the bisector parametrizations $\mathbf{B}_i(u,v), i = 1,2,3,4$ are rational. Furthermore, if $\sigma_s \in \mathbb{Q}[\delta](u,v)$, then $\mathbf{B}_i \in \mathbb{Q}[\delta](u,v), i = 1,2,3,4$.

Since the norm of the normal vector of all quadrics of revolution and tori is rational, we have:

Corollary 13.2 *The bisector of a circular-cylinder and a quadric of revolution or a torus sharing the same axis is rational.*

The Surface S Is Not of Revolution or It Is Not Sharing the Same Axis with the Cylinder C

In this case, after finding $t = t(u,v,s)$ from Eq. (13.20), it can be inferred, after some cumbersome calculations, that

$$F(u,v,s,t(u,v,s)) = f_1(u,v,\sigma_s,s)\, f_2(u,v,\sigma_s,s)$$

where $f_1(u, v, \sigma_s, s)$ and $f_2(u, v, \sigma_s, s)$ are quadratic in s.

Theorem 13.5 *Let C be a circular cylinder and S a torus or a quadric which is not of revolution, or it does not have the same axis of revolution as C. Let $\sigma_s(u, v) = \|\partial_u S \times \partial_v S\|$ be the norm of the normal vector of S. Then, there are four components of the parameterization for the untrimmed bisector $\mathbf{B}_i(u, v)$, $i = 1, 2, 3, 4$, that in general contain a square root.*

The proof of this theorem may be found in [1], Section 3.4.

Example 13.2.5 *Let*

$$C(s,t) = \begin{bmatrix} \dfrac{2s}{1+s^2} + 4 \\ t \\ \dfrac{1-s^2}{1+s^2} \end{bmatrix} \quad \textit{and} \quad S(u,v) = \begin{bmatrix} \dfrac{2u}{1+u^2+v^2} \\ \dfrac{v}{1+u^2+v^2} \\ \dfrac{2(1-u^2-v^2)}{1+u^2+v^2} \end{bmatrix}$$

be a circular cylinder and an ellipsoid, respectively. Note that the ellipsoid is not of revolution.

The four components of the bisector parametrization are given by:

$$B_i(u,v) = \begin{bmatrix} \dfrac{-2\,u\Big(4\left(s_i\left(1+u^2+v^2\right)\right)^2-4+3\,s_i\left(u^2-1+v^2\right)\Big)}{\left(1+u^2+v^2\right)\left(s_i\,u^2-2\,us_i{}^2+2\,u+s_i\,v^2-s_i\right)} \\ \dfrac{v\Big(\left(s_i\left(16\,u^2-6\,u+16\,v^2+16\right)\right)^2-1+15\,s_i\left(u^2-1+v^2\right)\Big)}{\left(1+u^2+v^2\right)\left(s_i\,u^2-2\,us_i{}^2+2\,u+s_i\,v^2-s_i\right)} \\ \dfrac{\Big(\left(s_i\left(u^2-1+v^2\right)\right)^2-1\Big)\left(2\,u^2+3\,u+2+2\,v^2\right)}{\left(1+u^2+v^2\right)\left(s_i\,u^2-2\,us_i{}^2+2\,u+s_i\,v^2-s_i\right)} \end{bmatrix},$$

$i = 1, 2, 3, 4$, *where*

$$s_1 = \frac{a_1 + b_1\sigma + r_1}{16c}, \quad s_2 = -\frac{a_2 + b_2\sigma + r_1}{16c}, \quad s_3 = \frac{a_3 + b_3\sigma + r_2}{16c},$$

$$s_4 = -\frac{a_4 + b_4\sigma + r_2}{16c}, \quad r_1 = \sqrt{\alpha_1 + \beta_1\,\sigma}, \quad r_2 = \sqrt{\alpha_2 + \beta_2\,\sigma},$$

$$\sigma = \sqrt{u^4 + 14\,u^2 + 2\,u^2v^2 + 62\,v^2 + 1 + v^4},$$

and $a_i, b_i, c, \alpha_i, \beta_i \in \mathbb{R}[u, v, \sigma]$ are of very large size.

The good component of the bisector (the set of points at equal distance from C and S) is shown in Fig. 13.9.

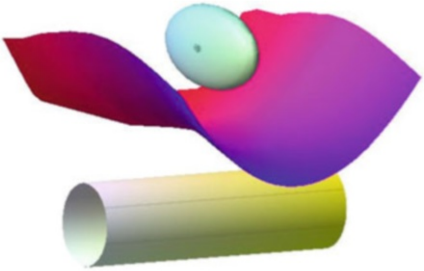


Fig. 13.9 Bisector of cylinder and ellipsoid

We will now show the results of applying the algorithm to parametrize the bisector of cylinders and cones.

Cylinder-Cylinder, Cylinder-Cone, or Cone-Cone Bisector

A cylindrical surface (or simply, cylinder) has a parametrization of the form

$$C(s,t) = \begin{bmatrix} f_1(s) + k_1 t \\ f_2(s) + k_2 t \\ f_3(s) + k_3 t \end{bmatrix},$$

where f_1, f_2, f_3, are rational functions, and k_1, k_2, k_3, are real constants. A conical surface (or simply, cone) has a parametrization of the form

$$\hat{C}(u,v) = \begin{bmatrix} v\, g_1(u) \\ v\, g_2(u) \\ v\, g_3(u) \end{bmatrix}$$

where g_1, g_2, g_3, are rational functions.

Applying the algorithm to compute the bisector of a cylinder and a cone, we get

$$G(u,v,s,t) = v\, G_0(u,v,s,t),$$

where $G_0(u,v,s,t)$ is linear in the parameters t and v. We obtain the following result, whose proof can be found in [1], Section 3.5.

Theorem 13.6 *Suppose S_1 and S_2 are respectively a cylinder and a cone. Let $\sigma_1(s,t) = \|\partial_s \mathbf{S}_1 \times \partial_t \mathbf{S}_1\|$ and $\sigma_2(u,v) = \|\partial_u \mathbf{S}_2 \times \partial_v \mathbf{S}_2\|$, be the norms of the respective normal vector of S_1 and S_2. Then*

1. *There exist two components of the parametrization for the untrimmed bisector $\mathbf{B}_i(s,u), i = 1,2$, that might contain a square root of a non-negative expression (originating from σ_1 and σ_2).*

2. *If* σ_1 *and* σ_2 *are rational, then the parametrization* $\mathbf{B}_i(s,u), i = 1, 2$, *is rational. Furthermore if* $\sigma_1 \in \mathbb{Q}[\delta_1](s,t), \sigma_2 \in \mathbb{Q}[\delta_2](u,v)$, *then* $\mathbf{B}_i \in \mathbb{Q}[\delta_1 \cdot \delta_2](s,u)$.

Example 13.2.6 *Let*

$$C_1(s,t) = \begin{bmatrix} s^2+2 \\ t \\ s \end{bmatrix} \quad \textit{and} \quad C_2(u,v) = \begin{bmatrix} \dfrac{u}{u^2+1} \\ \dfrac{1-u^2}{u^2+1} \\ v \end{bmatrix}$$

be a parabolic cylinder and an elliptic cylinder, respectively. The norms of the respective normal vectors are

$$\sigma_1(s,t) = \sqrt{1+4\,s^2} \;\; \textit{and} \;\; \sigma_2(u,v) = \frac{\sqrt{14\,u^2+1+u^4}}{\left(u^2+1\right)^2}.$$

The two components of the untrimmed bisector parametrization are given by:

$$B_1(u,s) = \begin{bmatrix} \dfrac{u\left(-16u\left(u^2+1\right)\left(1+4s^2\right)\left(s^2+2\right)+14\,u^2+1+u^4+4R\right)}{\left(u^2+8su-1\right)\left(-u^2+8su+1\right)\left(u^2+1\right)} \\ -\dfrac{\left(u^2-1\right)\left(-4u\left(u^2+1\right)\left(1+4s^2\right)\left(s^2+2\right)-2u^2\left(-1+24s^2\right)+u^4+1+R\right)}{\left(u^2+8su-1\right)\left(-u^2+8su+1\right)\left(u^2+1\right)} \\ -\dfrac{s\left(-\left(u^2+1\right)\left((u^4+1)(5+2s^2)+9u^2(6-4s^2)\right)+2u\left(14u^2+1+u^4\right)+8uR\right)}{\left(u^2+8su-1\right)\left(-u^2+8su+1\right)\left(u^2+1\right)} \end{bmatrix},$$

$$B_2(u,s) = \begin{bmatrix} -\dfrac{u\left(16\,u\left(u^2+1\right)\left(1+4\,s^2\right)\left(s^2+2\right)+u^4-14\,u^2-1+4\,R\right)}{\left(u^2+8\,su-1\right)\left(-u^2+8\,su+1\right)\left(u^2+1\right)} \\ \dfrac{\left(u^2-1\right)\left(4\,u\left(u^2+1\right)\left(1+4\,s^2\right)\left(s^2+2\right)-2\,u^2+48\,s^2u^2-u^4-1+R\right)}{\left(u^2+8\,su-1\right)\left(-u^2+8\,su+1\right)\left(u^2+1\right)} \\ \dfrac{s\left(u^2+1\right)\left(2\,u^4s^2+5\,u^4+32\,s^2u^2+54\,u^2-32\,s^2+5\right)+8\,uR}{\left(u^2+8\,su-1\right)\left(-u^2+8\,su+1\right)\left(u^2+1\right)} \end{bmatrix},$$

with $R = \left(s^2u^2+s^2+2\,u^2+2-u\right)\sqrt{(14\,u^2+1+u^4)\,(1+4\,s^2)}$. *See Fig. 13.10 for a plot of one component of the bisector.*

Remark 13.4 This approach can be applied in the same way to deal with the case of two cylinders or two cones.

13.3 Voronoï Diagram of Parallel Half-Lines Constrained to Compact Domain $\mathcal{D}_0 \subset \mathbb{R}^3$

Let $\{d_1, \ldots, d_n\}$ be a family of n parallel half-lines and $\mathcal{D}_0$ a compact domain of $\mathbb{R}^3$, defined, respectively, by (13.3) and (13.4). We are interested in computing the Voronoï diagram of $\{d_1, \ldots, d_n\}$ constrained to $\mathcal{D}_0$, with respect to the Euclidean distance.

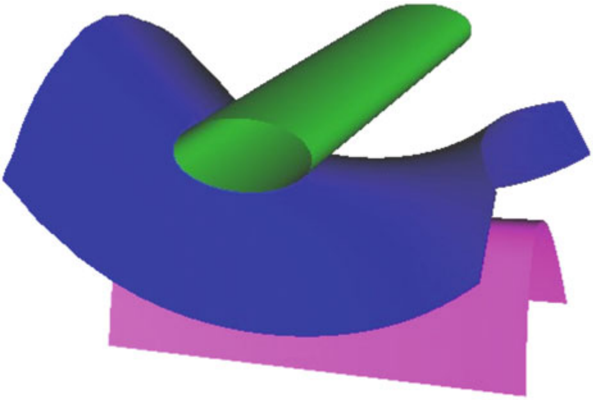


Fig. 13.10 Bisector of elliptic and parabolic cylinder

The Euclidean distance between a point $Q = (x, y, z)$ of $\mathbb{R}^3$ and a site d_i is given by:

$$\hat{g}_i(x, y, z) = \begin{cases} \mathrm{d}\left((x, y, z), (x_i, y_i, z_i)\right) & \text{if } x < x_i, \\ \mathrm{d}\left((y, z), (y_i, z_i)\right) & \text{if } x \geqslant x_i, \end{cases} \tag{13.21}$$

where $\mathrm{d}(\cdot, \cdot)$ denotes the Euclidean distance. The VD of $\{d_1, \ldots, d_n\}$ may be obtained as a minimization diagram (see [8, 14] and [58]), namely

$$VD(d_1, \ldots, d_n) = MD(\hat{g}_1, \ldots, \hat{g}_n), \tag{13.22}$$

and the Voronoi cell associated to a site d_i is

$$Vor(d_i) = \{Q \in \mathcal{D}_0 : \hat{g}_i(Q) \leqslant \hat{g}_j(Q), j = 1 \ldots n\}.$$

Furthermore, since $d(Q, d_j) < d(Q, d_k)$ if and only if $d^2(Q, d_j) < d^2(Q, d_k)$,

$$VD(d_1, \ldots, d_n) \equiv MD(g_1, \ldots, g_n),$$

where

$$g_i(x, y, z) = \begin{cases} \mathrm{d}^2\left((x, y, z), (x_i, y_i, z_i)\right) & \text{if } x < x_i, \\ \mathrm{d}^2\left((y, z), (y_i, z_i)\right) & \text{if } x \geqslant x_i, \end{cases} \tag{13.23}$$

Note that g_i is a piecewise polynomial expression in x, y, and z.

13.3.1 Bisectors, Trisectors and Quadrisectors of Sites

Let us present an algebraic description of the boundary components of a VD cell.

1. **Bisector of two sites:** Let d_i and d_j, $x_i < x_j$ be two sites with their respective associated distance functions g_i and g_j. Let $A_i = (x_i, y_i, z_i)$, $A_j = (x_j, y_j, z_j)$, and ℓ_i, ℓ_j be their respective endpoints and supporting lines.

   The bisector surface of d_i and d_j, is the equidistant surface from the two sites. It determines, locally, a VD face and it is defined by

$$b_{i,j} = \{(x, y, z) \in \mathbb{R}^3 : b_{i,j}(x, y, z) = 0\},$$

   where

$$b_{i,j}(x, y, z) = g_i(x, y, z) - g_j(x, y, z). \tag{13.24}$$

   The possible ambiguity in the notation will be clarified from the context. Expanding Eq. (13.24), we obtain:

$$b_{i,j}(x, y, z) = \begin{cases} p_{i,j}(x) + \ell_{i,j}(y, z) & \text{if } x < x_i, \\ q_{i,j}(x) + \ell_{i,j}(y, z) & \text{if } x_i \leqslant x < x_j, \\ \ell_{i,j}(y, z) & \text{if } x \geqslant x_j. \end{cases} \tag{13.25}$$

   where

$$p_{i,j}(x) = 2x(x_j - x_i) + x_i^2 - x_j^2, \tag{13.26}$$

$$q_{i,j}(x) = (x - x_j)^2, \tag{13.27}$$

$$\ell_{i,j}(y, z) = 2\,y(y_j - y_i) + {y_i}^2 - {y_j}^2 + 2\,z(z_j - z_i) + {z_i}^2 - {z_j}^2. \tag{13.28}$$

   Thus, the bisector surface consists of three connected pieces of surfaces:

   - One half plane (for $x < x_i$): contained in the bisector-plane of the two endpoints A_i and A_j;
   - One half parabolic-cylinder piece (for $x_i \leqslant x < x_j$): supported by the bisector-surface of the endpoint A_j and the supporting line ℓ_i (since $x_i < x_j$);
   - One half plane (for $x \geqslant x_j$): contained in the bisector-plane of the two supporting lines ℓ_i and ℓ_j.

2. **Trisector of three sites:** Let d_i, d_j and d_k, $x_i < x_j < x_k$, be three sites with their respective distance functions g_i, g_j and g_k. Let $A_i = (x_i, y_i, z_i)$, $A_j = (x_j, y_j, z_j)$, $A_k = (x_k, y_k, z_k)$, and ℓ_i, ℓ_j, ℓ_k be their respective endpoints and supporting lines.

   The trisector of the three sites d_i, d_j and d_k is the equidistant curve from the three sites. It determines, locally, a VD edge and it is given by the common intersection of the three ($3 = \binom{3}{2}$) bisectors spanned by three sites (2 by 2):

$$\mathcal{T}_{i,j,k} = \{(x,y,z) \in \mathbb{R}^3 : b_{i,j}(x,y,z) = b_{i,k}(x,y,z) = b_{j,k}(x,y,z) = 0\}. \tag{13.29}$$

Any equidistant portion of curve from more than three sites will be called a degenerate edge portion. This can happen, for example, if there are more than three neighbouring parallel half-lines which intersect a circle in the plane Oyz (see Figs. 13.16 and 13.17). The trisector $\mathcal{T}_{i,j,k}$ consists of four connected components of planar rational curves, and it is parametrized as follows:

$$\mathcal{T}_{i,j,k}(x) = \begin{cases} \left[x, -\frac{a_{0,1}+a_{1,1}x}{c_0}, \frac{a_{2,1}+a_{3,1}x}{c_0}\right]^T & \text{if } x < x_i, \\ \left[x, -\frac{a_{0,2}+a_{1,2}x+a_{2,2}x^2}{c_0}, \frac{a_{3,2}+a_{4,2}x+a_{5,2}x^2}{c_0}\right]^T & \text{if } x_i \leqslant x < x_j, \\ \left[x, \frac{a_{0,3}+a_{1,3}x+a_{2,3}x^2}{c_0}, -\frac{a_{3,3}+a_{4,3}x+a_{5,3}x^2}{c_0}\right]^T & \text{if } x_j \leqslant x < x_k, \\ \left[x, \frac{a_{0,4}}{c_0}, -\frac{a_{1,4}}{c_0}\right]^T & \text{if } x \geqslant x_k, \end{cases} \tag{13.30}$$

where: $c_0, a_{l,l'}$ are constant expressions depending on $x_i, y_i, z_i,\ y_j, z_j, z_j, x_k, y_k, z_k$. The four components of the trisector are:

- One half-line (for $x < x_i$) supported by the trisector line (the intersection of three bisector-planes) of the corresponding endpoints A_i, A_j and A_k,
- A piece of half-parabola (for $x_i \leqslant x < x_j$) supported by the intersection of the bisector-plane of the endpoints A_j and A_k, a parabolic-cylinder (the bisector of A_j and ℓ_i) and a parabolic-cylinder (the bisector of A_k and ℓ_i),
- A piece of half-parabola (for $x_j \leqslant x < x_k$) supported by the intersection of the bisector-plane of ℓ_i and ℓ_j, a parabolic-cylinder (the bisector of A_k and ℓ_i) and a parabolic-cylinder (the bisector of A_k and ℓ_j),
- One half-line parallel to the $x-$direction (for $x \geqslant x_k$) supported by the trisector line (the intersection of the three bisector-planes) of the corresponding lines ℓ_i, ℓ_j and ℓ_k.

Note that although each piece is a planar curve, they are not contained in the same plane.

3. **Quadrisector of four sites:** The quadrisector is the equidistant point from four sites $d_i, d_j, d_k,\ and\ d_l$. It determines the VD vertex and it is defined by the common point of four ($4 = \binom{4}{3}$) trisectors spanned by the four sites (3 by 3):

$$\mathcal{Q}_{i,j,k,l} = \mathcal{T}_{i,j,k} \cap \mathcal{T}_{i,j,l} \cap \mathcal{T}_{i,l,k} \cap \mathcal{T}_{j,l,k} \tag{13.31}$$

Any equidistant point from more than four sites is called a degenerate vertex.

Let us introduce some definitions and the regularity criteria for the topology of algebraic curves and surfaces introduced by L. Alberti and B. Mourrain [2] and

Liang et al. [43], which can be applied to compute the topology of the trisector curves and the bisector surfaces in a compact domain.

Definition 13.2 Let $\{\mathrm{g}_1, \ldots, \mathrm{g}_n\}$ be a family of distance functions associated to a family of sites $\{d_1, \ldots, d_n\}$ and a compact domain $\mathcal{D} = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$.

1. A distance g_{i_0} is said to be `active` in $\mathcal{D}$ if it contributes, locally, to the lower envelope of the family of distance functions $\{\mathrm{g}_1, \ldots, \mathrm{g}_n\}$,

$$\text{i.e. : } \min(\mathrm{g}_{i_0|_{\mathcal{D}}}) \leqslant \min\{\max(\mathrm{g}_{i|_{\mathcal{D}}}), i = 1 \ldots, n\},$$

and the site d_{i_0} is said to be `active` in the domain $\mathcal{D}$.
2. For a domain $\mathcal{D} \subset \mathbb{R}^3$, we call d-list a list of distance functions restricted to $\mathcal{D}$, denoted by $B(\mathcal{D}) = \{\mathrm{g}_{1|_{\mathcal{D}}}, \ldots, \mathrm{g}_{p|_{\mathcal{D}}}\}$, so $|B(\mathcal{D})| = p$.

Definition 13.3 ([2,43]) An algebraic curve $\mathcal{C}$ (or surface $\mathcal{S}$) is said to be t-regular in a compact domain $\mathcal{D}$ if its topology is uniquely determined from its intersection with the boundary of $\mathcal{D}$.

A domain $\mathcal{D}$ is said to be t-regular if the bisectors and trisectors within it are t-regular.

Definition 13.4 A d-list $B(\mathcal{D})$ is said to be b-regular if the corresponding bisectors and trisectors are t-regular in $\mathcal{D}$.

Since in our context we are interested in computing the topology of VD faces (bisectors), edges (trisectors) and vertices (quadrisectors) in a domain $\mathcal{D}$, any d-list with more than four distance functions is not b-regular, whence the b-regularity will be effectively checked for a d-list containing at most four distance functions as follows:

1. A d-list $B(\mathcal{D})$ with one active distance function $\mathrm{g}_{i|_{\mathcal{D}}}$ ($\mathcal{D}$ does not contain a VD cell, i.e. $\mathcal{D} \subset Vor(d_i)$) is trivially `b-regular`.
2. A d-list $B(\mathcal{D})$ with two active distance functions $\mathrm{g}_{i|_{\mathcal{D}}}$ and $\mathrm{g}_{j|_{\mathcal{D}}}$ ($\mathcal{D}$ might intersect one bisector surface) is `b-regular` if the bisector $b_{i,j}$ is `t-regular` in the domain $\mathcal{D}$.
3. A d-list $B(\mathcal{D})$ with three active distance functions $\mathrm{g}_{i|_{\mathcal{D}}}$, $\mathrm{g}_{j|_{\mathcal{D}}}$ and $\mathrm{g}_{k|_{\mathcal{D}}}$ ($\mathcal{D}$ might intersect three bisectors and one trisector) is `b-regular` if:
   (a) Each one of the three bisectors $b_{i,j}$, $b_{i,k}$ and $b_{j,k}$, separately, is `t-regular` in $\mathcal{D}$;
   (b) The trisector curve $\mathcal{T}_{i,j,k}$ is `t-regular` in $\mathcal{D}$.
4. A d-list $B(\mathcal{D})$ with four active distance functions $\mathrm{g}_{i|_{\mathcal{D}}}$, $\mathrm{g}_{j|_{\mathcal{D}}}$, $\mathrm{g}_{k|_{\mathcal{D}}}$ and $\mathrm{g}_{l|_{\mathcal{D}}}$ (thus, $\mathcal{D}$ might intersect six bisectors, four trisectors and one quadrisector) is `b-regular` if:
   (a) Each one of the six bisectors $b_{m,n}$, $m < n \in \{i, j, k, l\}$, separately, is `t-regular` in $\mathcal{D}$,

(b) Each one of the four trisectors $\mathcal{T}_{m,n,p}$, $m < n < p \in \{i, j, k, l\}$, separately, is `t-regular` in $\mathcal{D}$, and

(c) The corresponding quadrisector point $Q_{i,j,k,l}$ is contained in $\mathcal{D}$.

In the next section, we describe the main steps of the algorithm for computing the VD.

13.3.2 The Algorithm

The approach consists of two main parts: the subdivision phase of the initial domain $\mathcal{D}_0$, and the reconstruction phase of polyhedra isotopic to the VD cells constrained to $\mathcal{D}_0$:

- During the subdivision phase, the initial domain $\mathcal{D}_0$ will be subdivided into sub-domains which are t-regular or have size smaller than ϵ. If $\mathcal{D}_0$ is not t-regular, it is divided in two sub-domains by a mediator plane which is perpendicular to one of the directions of the coordinate system. This process is repeated with each sub-domain, and so on, following a kd-tree structure (see [6]), until all sub-domains are t-regular or have size smaller than ϵ. The sub-domains that intersect VD cells will be identified and connected following an adjacency graph. In the subdivision phase, floating-point computations are done using interval arithmetic, a well known technique to reduce accumulated rounding errors (see [3, 9, 47]). In interval arithmetic the real numbers are represented by intervals whose endpoints are floating-point numbers. Interval arithmetic has been used often in Computational Geometry and CAGD to design robust algorithms (see [48, 54]). For example, a combination of a subdivision method and interval arithmetic has been applied in [33] to the computation of arrangements.
- The reconstitution phase consists in traversing the adjacency graph using a depth-first search (**DFS**) algorithm (see [10, 21, 28, 39, 55] and [35]), while meshing all bisector surfaces and trisector curves in each traversed sub-domain.

Subdivision Phase

Let $\mathcal{D}_0 = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ be the initial domain, the n sites $d_1, \ldots d_n$, and a threshold $\varepsilon > 0$.

1. The process begin by computing the initial d-list

$$B(\mathcal{D}_0) := \{g_{1|_{\mathcal{D}_0}}, \ldots, g_{n|_{\mathcal{D}_0}}\}$$

carrying the distance functions associated to the n sites.

2. Then we initialize:

   - A stack $\mathcal{P}$ by the d-list $B(\mathcal{D}_0)$ and
   - The adjacency graph $\mathcal{G}$ by the empty graph.

3. The principal step starts with a loop by taking a d-list $B(\mathcal{D})$ from the stack $\mathcal{P}$. Using interval arithmetic, the distance functions are bounded on the sub-domain $\mathcal{D}$, i.e.,

$$m^i_{\mathcal{D}} \leqslant g_{i_{|\mathcal{D}}} \leqslant M^i_{\mathcal{D}}, i = 1, \ldots, |B(\mathcal{D})|.$$

All distance functions which are not active in $\mathcal{D}$ are filtered out, i.e., all distances $g_{i_0{}_{|\mathcal{D}}}$ verifying

$$m^{i_0}_{\mathcal{D}} \geqslant \min\{M^i_{\mathcal{D}}, i = 1, \ldots, |B(\mathcal{D})|\},$$

will be eliminated from the d-list $B(\mathcal{D})$:

$$B(\mathcal{D}) := B(\mathcal{D}) \diagdown \{g_{i_0{}_{|\mathcal{D}}}\}.$$

The resulting d-list, which contains only the active distances, will be tested for regularity:

- If the d-list $B(\mathcal{D})$ is b-`regular` or if the threshold size on the sub-domain is reached (i.e.: $|\mathcal{D}| = \max_{1 \leqslant i \leqslant 3}(|b_i - a_i|) < \varepsilon$), then a list of the corresponding bisectors and trisectors will be computed:

$$B^{bis}(\mathcal{D}) := \left\{\mathcal{T}_{i,j,k} = b_{i,j} \cap b_{i,k} \cap b_{j,k}, b_{i,j}, i < j < k \in \{1, \ldots, |B(\mathcal{D})|\}\right\},$$

  and it will be placed in the graph $\mathcal{G}$.
- Else, the domain $\mathcal{D}$ will be subdivided in two sub-domains $\mathcal{D}_1$ and $\mathcal{D}_2$ by a mediator plane which is perpendicular to the direction of the longest edge of $\mathcal{D}$. The adjacency graph $\mathcal{G}$ will be updated by replacing the d-list $B(\mathcal{D})$ by the two new corresponding d-lists $B(\mathcal{D}_1)$ and $B(\mathcal{D}_2)$, by adding an adjacency edge between them, and connecting them with their neighbour d-lists. Finally, they are stored in the stack $\mathcal{P}$.

  The process continues until the stack $\mathcal{P}$ is empty.

However, the d-lists encountered when the threshold size is reached might span degenerate VD edges and/or degenerate VD vertices. Indeed, this could happen if all involved bisectors (and/or trisector) have some common intersection curve portion (resp. and/or point) in the corresponding sub-domain.

At the end of the process, we obtain an adjacency graph in which the d-lists of bisectors and trisectors are the nodes (or vertices) and the adjacencies (between lists) are the edges. These nodes (lists $B^{bis}(\mathcal{D})$) could span: (1) one bisector, (2) three bisectors and one trisector, (3) six bisectors, four trisectors and one quadrisector, or eventually (4) degenerate VD edges and degenerate VD vertices, in the domain $\mathcal{D}$. The adjacency graph will serve for the traversal path during the next phase.

Reconstruction Phase

This part of the method consists in the reconstruction of the VD cells, by traversing and meshing all lists $B^{bis}(\mathcal{D})$ sorted in the adjacency graph $\mathcal{G}$.

- The traversal of the lists will be realized by applying a **DFS** algorithm (see [28] and [35]). The **DFS** algorithm is a way of traversing a graph or tree $\mathcal{G}$, starting at one specific node and going through all nodes exactly once. The algorithm is successively applied to each sub-graph $\mathcal{G}_{l,k}$ of the adjacency graph $\mathcal{G}$, where $\mathcal{G}_{l,k}$ is the sub-graph of all lists containing the bisector $b_{l,k}$:

$$\mathcal{G}_{l,k} = \{B^{bis}(\mathcal{D}) \in \mathcal{G} : b_{l,k} \in B^{bis}(\mathcal{D})\}.$$

- The meshing process of a list consists in constructing a polyhedron whose topology is isotopic to the parts of the bisectors and trisectors that compose the VD faces and VD edges, respectively, in the corresponding domain. Their topology is determined from their intersection points with the boundary of the domain (see the Figs. 13.11–13.15).

We introduce the following definition.

Definition 13.5 A site d_p is said to be `involved` in a list $B^{bis}(\mathcal{D})$, if there exists $q \in \{1, \dots, |B(\mathcal{D})|\}$ such that $b_{p,q} \in B^{bis}(\mathcal{D})$.

An approximation of the boundary of the VD cell associated to a site d_i, $\mathcal{VD}[i]$, will be reconstructed by traversing and meshing all lists in which the site is involved. All open VD cells will be completed by a piece of the boundary $\partial\mathcal{D}_0$ such that the reconstruction will be constrained to $\mathcal{D}_0$.

The process of the reconstruction starts from a list $B^{bis}(\mathcal{D})$ (a node) of $\mathcal{G}$, which has not been traversed. The boundary of $\mathcal{VD}[i]$, will be obtained as follows:

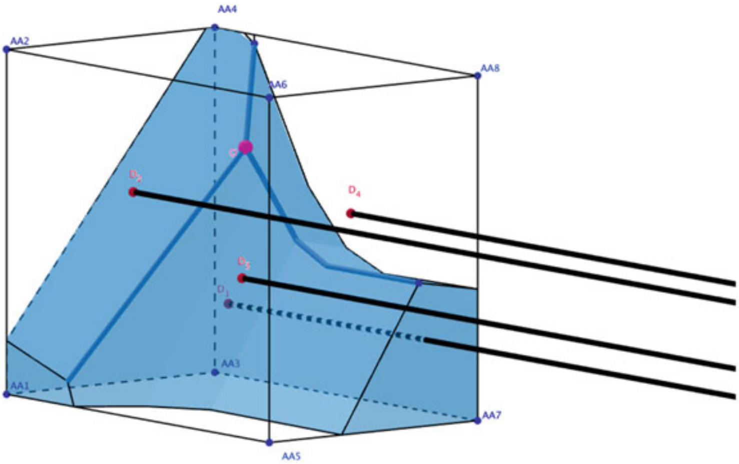


Fig. 13.11 Cell associated to d_1

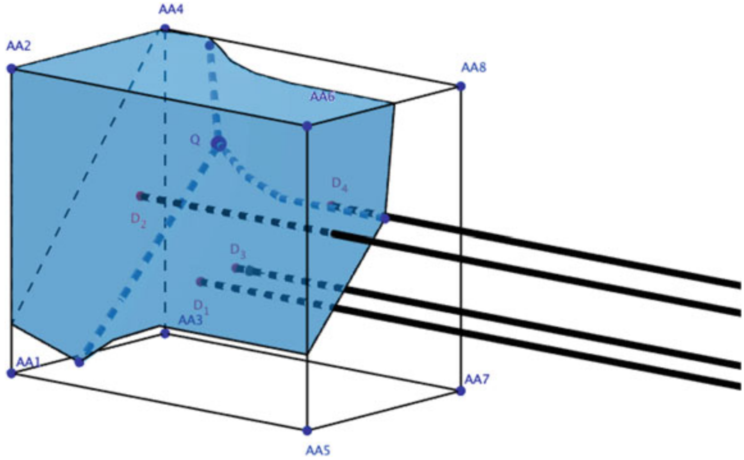


Fig. 13.12 Cell associated to d_2

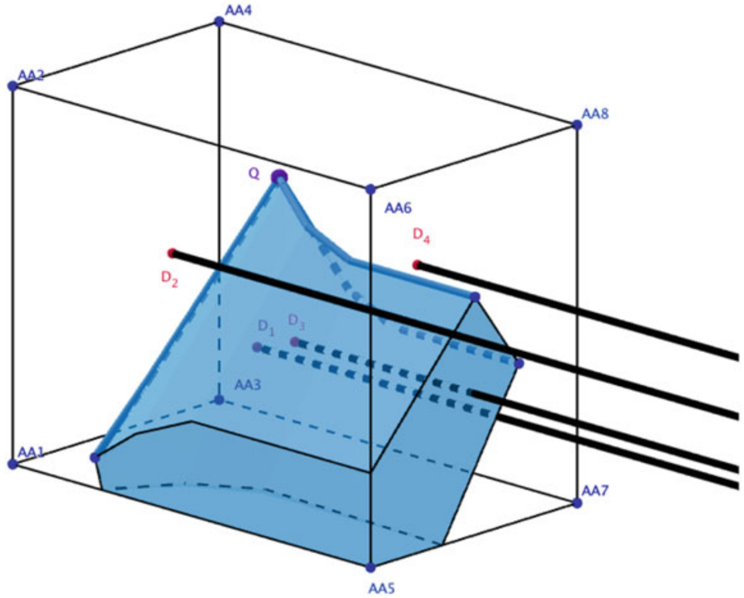


Fig. 13.13 Cell associated to d_3

1. For all sites d_j involved in $B^{bis}(\mathcal{D})$, $j \neq i$, follow the sub-graph $\mathcal{G}_{i,j}$ of all lists that contain the bisector $b_{i,j}$, starting from $B^{bis}(\mathcal{D})$ by $\mathbf{DFS}(\mathcal{G}_{i,j}, B^{bis}(\mathcal{D}))$, and mesh the bisectors and trisectors of all traversed lists.
2. If, during the traversal, we reach a list $B^{bis}(\mathcal{D})_1$ that contains a VD edge $\mathcal{T}_{i,j,j_1}$, then the site d_j will be substituted by a site d_{j_1}, $j_1 \neq i$ and continue the traversal with the sub-graph $\mathcal{G}_{i,j_1} \subset \mathcal{G}$, of all lists that contain the bisector b_{i,j_1} starting

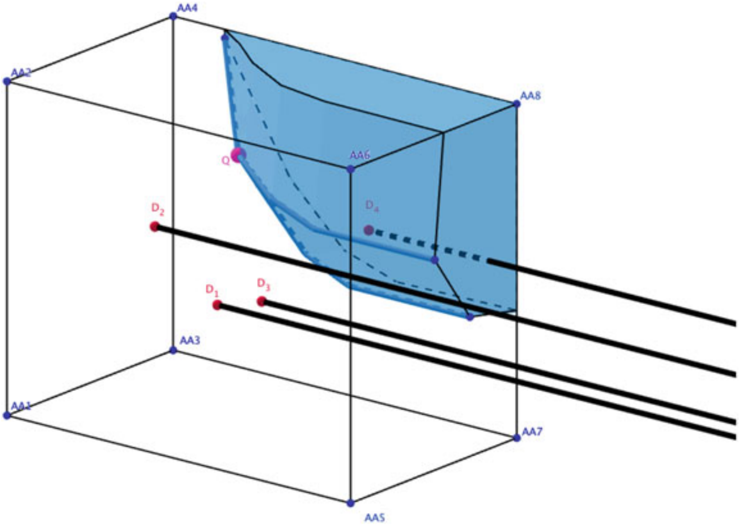


Fig. 13.14 Cell associated to d_4

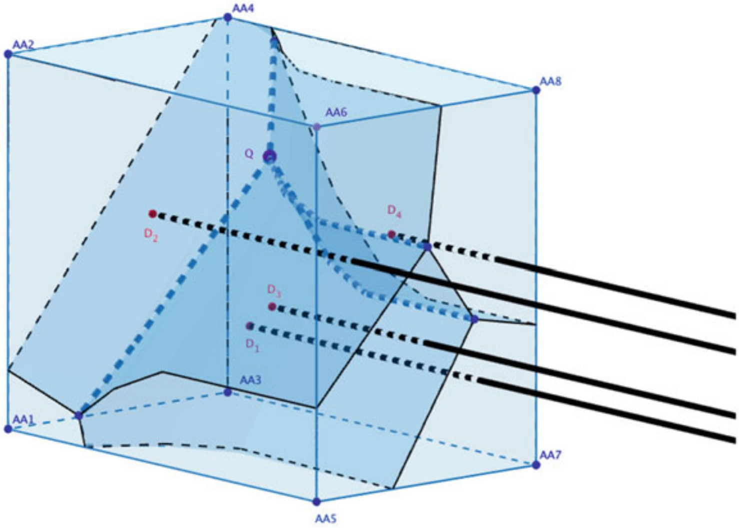


Fig. 13.15 VD cell four site d_1, d_2, d_3 and d_4

from $B^{bis}(\mathcal{D})_1$ by: **DFS**$(\mathcal{G}_{i,j_1}, B^{bis}(\mathcal{D})_1)$, and mesh the bisectors and trisectors of all traversed lists, as before.

3. The process continues until reaching the initial list $B^{bis}(\mathcal{D})$ for the second time.

The VD cell $\mathcal{VD}[i]$ is therefore covered. We repeat the same process for all sites. At the end we obtain a polyhedral approximation of each VD cell, which is isotopic to the VD cell in the domain $\mathcal{D}_0$.

Remark 13.5 (a) The approximate VD computed is topologically equivalent to the exact VD on each sub-domain $\mathcal{D}$ such that $|\mathcal{D}| \geqslant \varepsilon$, or when $|\mathcal{D}| < \varepsilon$ and the corresponding list $B^{bis}(\mathcal{D})$ contains only one VD edge or only one VD vertex. In other cases the topology is unknown.

(b) The same process for computing the topology of VD faces and VD edges in the regular case can be used in the case of degenerate VD edges and degenerate VD vertices.

The implementation of the algorithm is being developed, but has not been completed yet. Some subroutines has been tested with satisfactory results. To illustrate the approach, we present a prototype example with four sites, using Maple software and GeoGebra 3D.

Example 13.3.1 *Let* $d_1 = (-5+t, 3, -2)$*,* $d_2 = (-3+t, -4, 4)$*,* $d_3 = (3+t, 4, 3)$ *and* $d_4 = (t, 0, 0)$ *be four half-lines and the domain* $\mathcal{D}_0 = [-10, 10] \times [-6, 6] \times [-6, 10]$.

The example requires the computation of about 45 intersection points (bisector surface and trisector curve with the border of sub-domains), and building and connecting about 65 triangles. The meshed VD cells of $\{d_1,\ d_2,\ d_3,\ d_4\}$ *constrained to* $\mathcal{D}_0$ *are separately shown in the Figs. 13.11–13.14, and the four VD cells are simultaneously presented in Fig. 13.15.*

Remark 13.6 Suppose we have five parallel half-lines, which intersect a circle $\mathcal{C}$ in the plane (Oyz). Then we obtain a portion of degenerate VD edge as announced in the trisector definition 2. More precisely, using the notation in Eq. 13.3, let d_1, d_2, d_3, d_4, d_5 be parallel half-lines, which intersect $\mathcal{C}$, such that $x_1 < x_2 < x_3 < x_4 < x_5$. In this situation, we obtain a portion of degenerate VD edge: a half-line with endpoint A, whose first coordinate is equal to x_5, parallel to the given d_i, and passing through the center of $\mathcal{C}$ (see Fig. 13.16). For $x < x_5$ the edge bifurcates in five branches of curve, all starting from A. Each branch consists of two pieces of

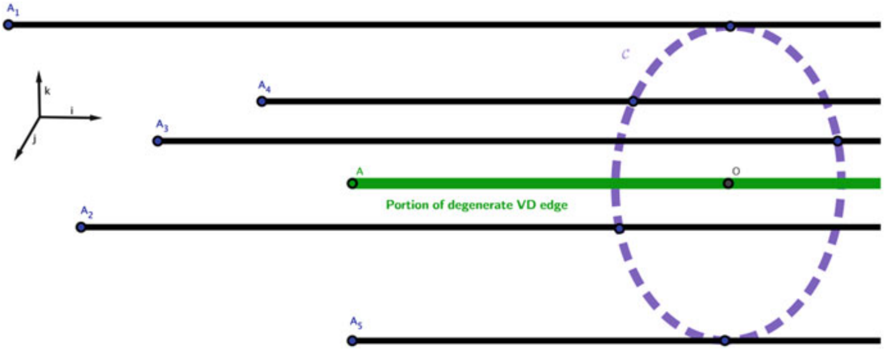


Fig. 13.16 The portion of degenerate edge (equidistant from the five half-lines) in *green*

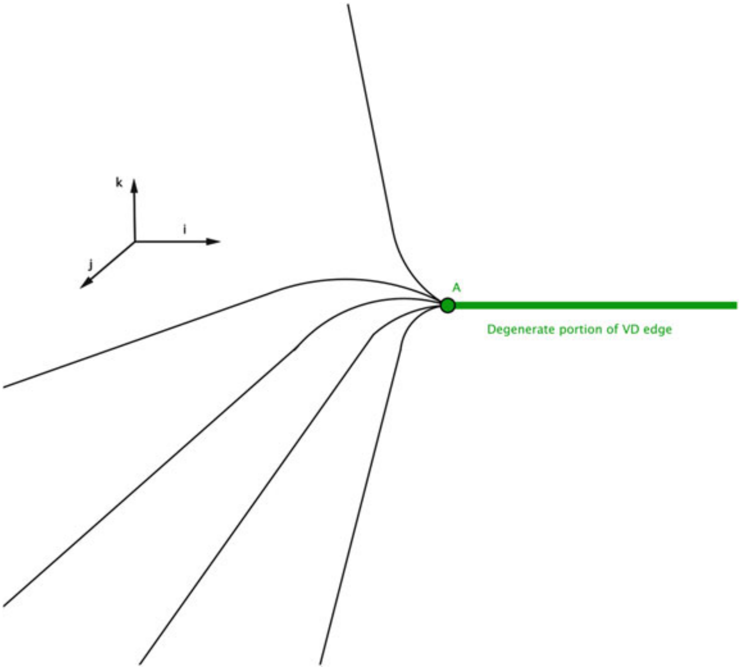


Fig. 13.17 The five branches of VD edge (for $x < x_5$) started at end point A of the portion of degenerate VD edge (for $x \geqslant x_5$)

half-parabola and one half-line (see Fig. 13.17). The algorithm we present can deal easily with this case.

Conclusion

We presented new approaches for computing an exact algebraic parametrization of a planar curve-curve bisector and two low degree rational surfaces bisector, and an automatic geometric and numerical characterization of the planar point-curve and curve-curve bisector. In addition, we presented a new algorithm for computing a Voronoï diagram of a set of parallel half-lines in tri-dimensional space, constrained to a compact domain.

1. For the algebraic parametrization of bisectors, in most cases involving two low degree planar rational curves or rational surfaces, the new methods proposed produce, efficiently, an exact parametrization of the bisector. This includes line-conic/cubic, circle-conic/cubic, circle-curve, plane-quadric, plane-torus, circular cylinder-quadric, circular cylinder-torus, cylinder-cylinder, cylinder-cone, and cone-cone bisectors.

(continued)

Even if the parameterization obtained is not rational in general, we get a representation that can be used easily for approximation purposes and for the trimming process in the planar case.

2. For an automatic geometric and numerical characterization for the planar point-curve and curve-curve true bisector, using color model **RGB** in GeoGebra, the true bisector points are scanned and displayed in white color. Although the method does not provide an algebraic representation, it allows the collection of the coordinates of a dense set of bisector points, up to a fixed small error, that can be used to compute the approximate representation of the bisector. It should be noted that the approach is similar to that introduced by Shou et al. [59], but much more advantageous by obtaining an effective list of coordinates points of the true bisector, not only for the point-curve case, but also for the curve-curve case.
3. For the VD, based on subdivision process and meshing algebraic curve and surface in a compact domain, we presented a new algorithmic approach for computing an approximative VD of new kind of sites, the parallel half-lines, constrained to a compact domain $\mathcal{D}_0 \subset \mathbb{R}^3$, and with respect to the Euclidean distance. Unlike some approximative algorithms, the algorithm we presented computes an approximation of the VD, which is topologically equivalent to the exact one, for all sub-domains $\mathcal{D}$ with $|\mathcal{D}| \geqslant \varepsilon$, and when $|\mathcal{D}| < \varepsilon$ and the corresponding list $B^{bis}(\mathcal{D})$ contains only one VD edge or only one VD vertex. In other cases, an advanced method have to be developed for guarantying correct topology of the VD. The approach could be applied for a compact domain of any metric space, in which, the distance function is transformable in a polynomial expression. We plan to implement our algorithm in a programming language for an effective analysis of the approach. In the future, we plan to extend the approach to the case of non-Euclidean distance and non parallel half-lines, which could be a more complicated problem.

Acknowledgements The authors want to thank the reviewers for the appropriate suggestions that have helped to improve the paper.

References

1. I. Adamou, Curve and surface bisectors, and Voronoi diagram of a family of parallel half-lines in $\mathbf{R}^3$, Ph.D. thesis, Universidad de Cantabria (2013). http://repositorio.unican.es/xmlui/handle/10902/3699
2. L. Alberti, B. Mourrain, Regularity criteria for the topology of algebraic curves and surfaces, in *Proceedings of IMA International Conference on Mathematics of Surfaces XII*, ed. by R. Martin, M. Sabin, J. Winkler (Springer, Berlin/Heildelbeg, 2007), pp. 1–28

3. G. Alefeld, J. Herzberger, *Introduction to Interval Computations*. Computer Science and Applied Mathematics (Academic, New York, 1983)
4. F. Aurenhammer, R. Klein, Voronoi diagrams, in *Handbook of Computational Geometry*, J.-R. Sack, J. Urrutia (North-Holland, Amsterdam, 2000), pp. 201–290
5. P.G. Barros, D.A. Pessoa, P.J.S. Leite, V. Teichrieb, J. Kelner, Three dimensional oil well planning in ultra-deep water, in *Proceedings of Symposium on Virtual Reality (SVR)*, Belém, vol. 1, 2006, pp. 285–296
6. J.L. Bentley, Multidimensional binary search trees used for associative searching. Commun. ACM **18**, 509–517 (1975)
7. I. Boada, N. Coll, N. Madern, J.A. Sellares, Approximations of 2d and 3d generalized Voronoi diagrams. Int. J. Comput. Math. **85**(7), 1003–1022 (2008)
8. J.-D. Boissonnat, C. Wormser, M. Yvinec, Curved Voronoi diagrams, in *Effective Computational Geometry for Curves and Surfaces*, ed. by J.-D. Boissonnat, M. Teillaud (Springer, Berlin/Heidelberg, 2006), pp. 67–116
9. H. Brönnimann, C. Burnikel, S. Pion, Interval arithmetic yields efficient dynamic filters for computational geometry. Discret. Appl. Math. **109**, 25–47 (2001)
10. T.H. Cormen, C.E. Leiserson, R.L. Rivest, S. Clifford, *Introduction to Algorithms*, 2nd edn. (MIT/McGraw-Hill, 2001), pp. 540–549. ISBN:0-262-03293-7. Section 22.3: Depth-first search
11. H.P. Decell, An application of the Cayley-Hamilton theorem to generalized matrix inversion. SIAM Rev. **7**, 526–528 (1965)
12. G. Diaz-Toca, L. Gonzalez-Vega, H. Lombardi, Generalizing Cramer's rule: solving uniformly linear systems of equations. SIAM J. Matrix Anal. Appl. **27**(3), 621–637 (2005)
13. D. Dutta, C. Hoffman, On the skeleton of simple CSG objects. ASME J. Mech. Des. **115**, 87–94 (1993)
14. H. Edelsbrunner, R. Seidel, Voronoi diagrams and arrangements. Discret. Comput. Geom. **1**, 25–44 (1946)
15. G. Elber, M.-S. Kim, Bisector curves of planar rational curves. Comput. Aided Des. **30**(14), 1089–1096 (1998)
16. G. Elber, M.-S. Kim, Computing rational bisectors. IEEE Comput. Graph. Appl. **19**(6), 76–81 (1999)
17. G. Elber, M.-S. Kim, Rational bisectors of CSG primitives, in *Proceedings of the 5th ACM/IEEE Symposium on Solid Modeling and Applications*, Ann Arbor, June 1999, pp. 246–257
18. G. Elber, M.-S. Kim, A computational model for non-rational bisector surfaces: curve-surface and surface-surface bisectors, in *Proceedings of the Geometric Modeling and Processing 2000*, Hong Kong, Apr 2000, pp. 364–372
19. I.Z. Emiris, A. Mantzaflaris, B. Mourrain, Voronoi diagrams of algebraic distance fields. Compu. Aided Design. **45**, 511–516 (2013)
20. I.Z. Emiris, E.P. Tsigaridas, G.M. Tzoumas, Exact Delaunay graph of smooth convex pseudo-circles: general predicates, and implementation for ellipses, in *SIAM/ACM Joint Conference on Geometric and Physical Modeling*, San Francisco, Oct 2009
21. S. Even, *Graph Algorithms* (Cambridge University Press, 2011), pp. 46–48. ISBN:978-0-521-73653-4
22. H. Everett, C. Gillot, D. Lazard, S. Lazard, M. Pouget, The Voronoi diagram of three arbitrary lines in $\mathbb{R}^3$, in *Abstracts of 25th European Workshop on Computational Geometry*, Brussels, 2009
23. C. Fan, J. Luo, J. Liu, Y. Xu, Half-plane Voronoi diagram, in *Proceedings of the 2011 Eighth International Symposium on Voronoi Diagrams in Science and Engineering*, Qingdoa, pp. 127–133
24. R.T. Farouki, J.K. Johnstone, The bisector of a point and a plane parametric curve. Comput. Aided Geom. Des. **11**(2), 117–151 (1994)

25. R.T. Farouki, J.K. Johnstone, Computing point/curve and curve/curve bisectors, in *Design and Application of Curves and Surfaces: Mathematics of Surfaces V*, ed. by R.B. Fisher (Oxford University Press/Clarendon Press, 1994), pp. 327–354
26. R.T. Farouki, R. Ramamurthy, Specified-precision computation of curve/curve bisectors. Int. J. Comput. Geom. Appl. 8(5–6), 599–617 (1998)
27. S. Fortune, Voronoi diagrams and Delaunay triangulations, in *Handbook of Discrete and Computational Geometry*, ed. by J.E. Goodman, J. O'Rourke (CRC, Boca Raton, 1997), pp. 377–388
28. H. de Fraysseix, P. Ossona de Mendez, P. Rosenstiehl, Trémaux trees and planarity. Int. J. Found. Comput. Sci. **17**(5), 1017–1030 (2006). doi:10.1142/S0129054106004248
29. J.T. Fredrich, D. Coblentz, A.F. Fossum, B.J. Thorne, Stress perturbations adjacent to salt bodies in the deepwater Gulf of Mexico, in *SPE Annual Technical Conference and Exhibition*, Denver, 5–8 Oct 2003
30. F. Gerritsen, Evolution in color (Schiffer Publishing, West Chester). Originally published in Dutch in 1982 as Evolutie van de Kleurenleer. Translation by Dr. Edward Force and Ruth de Vriendt, 1988
31. I. Hanniel, G. Elber, Computing the Voronoi cells of planes, spheres and cylinders in $\mathbb{R}^3$. Comput. Aided Geom. Des. **26**(6), 695–710 (2009)
32. M. Hasan, M.L. Gavrilova, A geometric approach to drill path collision avoidance, in *Proceedings of the 2010 International Symposium on Voronoi Diagrams in Science and Engineering*, Quebec, 2010, pp. 244–253
33. Y. Hijazi, T. Breuel, Computing arrangements using subdivision and interval arithmetic, in *Proceedings of the Sixth International Conference on Curves and Surfaces*, Avignon, 2006, pp. 173–182
34. K.E. Hoff, J. Keyser, M. Lin, D. Manocha, T. Culver, Fast computation of generalized Voronoi diagrams using graphics hardware, in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, July 1999, pp. 277–286
35. J. Hopcroft, R.E. Tarjan, Efficient planarity testing. J. Assoc. Comput. Mach. **21**(4), 549–568 (1974). doi:10.1145/321850.321852
36. M.-S. Kim, G. Elber, J.-K. Seong, Geometric computations in parametric space, in *Spring Conference on Computer Graphics Bundmerice Castle*, Slovak Republic, May 2005
37. D. Kim, D.-S. Kim, Region-expansion for the Voronoi diagram of 3D spheres. Comput. Aided Des. **38**, 417–430 (2006)
38. R. Klein, in *Concrete and Abstract Voronoi Diagrams*, ed. by G. Goos, J. Hartmanis. Lecture Notes in Computer Science (Springer, Berlin/New York, 1989)
39. D.E. Knuth, *The Art of Computer Programming*, vol. 1, 3rd edn. (Addison-Wesley, 1997). ISBN:0-201-89683-4, OCLC 155842391, www.worldcat.org/oclc/155842391
40. V. Koltun, M. Sharir, Three dimensional euclidean Voronoi diagrams of lines with a fixed number of orientations. SIAM J. Comput. **32**(3), 616–642 (2003)
41. J.C. Lafon, G. Blanc, Meshing and visualization of oilfields, in *IEEE Conference on Information Visualization (IV '97)*, London, 1997
42. H. Ledoux, Computing the 3D Voronoi diagram robustly: an easy explanation, in *4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007)*, Pontypridd, 2007
43. C. Liang, B. Mourrain, J.P. Pavone, Subdivision methods for 2d and 3d implicit curves, in *Geometric Modeling and Algebraic Geometry*, ed. by B. Jüttler, R. Piene (Springer, Berlin/Heidelberg, 2008), pp. 199–214
44. R. Losada, T. Recio, J.L. Valcarce, Equal bisectors at a vertex of a triangle, in *ICCSA 2011*, ed. by B. Murgante et al. Part IV. LNCS, vol. 6785 (Springer, Berlin/Heidelberg, 2011), pp. 328–341
45. W. Lü, Rationality of the offsets to algebraic curves and surfaces. Appl. Math. J. Chin. Univ. **9**(3), 265–278 (1994)
46. W. Lü, Rational parametrization of quadrics and their offsets. Computing **57**(2), 135–147 (1996)

47. R.E. Moore, *Interval Analysis* (Prentice Hall, Englewood Cliffs, 1966)
48. B. Mourrain, S. Pion, S. Schmitt, J.-P. Técourt, E. Tsigaridas, N. Wolpert, Algebraic issues in computational geometry, in *Effective Computational Geometry for Curves and Surfaces*, ed. by J.-D. Boissonnat, M. Teillaud (Springer, Berlin/Heidelberg, 2006), pp. 117–155
49. R.B. Norman, *Electronic Color: The Art of Color Applied to Graphic Computing* (Van Nostrand Reinhold, New York, 1990)
50. A. Okabe, B. Boots, K. Sugihara, *Spatial Tessellations, Concepts and Applications of Voronoi Diagrams* (Wiley, New York, 2000)
51. M. Peternell, Geometric properties of bisector surfaces. Graph. Models **62**(3), 202–236 (2000)
52. M. Peternell, Sphere-geometric aspects of bisector surfaces, in *AGGM 2006*, Barcelona, 2006
53. H. Pottmann, Rational curves and surfaces with rational offsets. Comput. Aided Geom. Des. **12**(2), 175–192 (1995)
54. S. Schirra, Robustness and precision issues in geometric computation, in *Handbook of Computational Geometry*, ed. by J.-R. Sack, J. Urrutia (North-Holland, Amsterdam, 2000), pp. 597–632
55. R. Sedgewick, *Algorithms in C++: Graph Algorithms*, 3rd edn. (Pearson, 2002). ISBN:978-0-201-36118-6
56. J.K. Seong, E. Cohen, G. Elber, Voronoi diagram computations for planar NURBS curves, in *Proceedings of the 2008 ACM Symposium on Solid and Physical Modelling*, Stony Brook, 2–4 June 2008
57. O. Setter, M. Sharir, D. Halperin, Constructing two-dimensional Voronoï diagrams via divide-and-conquer of envelopes in space, in *Transactions on Computational Science IX*, ed. by M.L. Gavrilova, C.J.K. Tan (Springer, Berlin/Heidelberg, 2010), pp. 1–27
58. M. Sharir, Almost tight upper bounds for lower envelopes in higher dimensions. Discret. Comput. Geom. **12**, 327–345 (1994)
59. H. Shou, T. Li, Y. Miao, The bisector of a point and a plane algebraic curve, in *Theoretical and Mathematical Foundations of Computer Science*, Singapore, ed. by Q. Zhou, vol. 164 (Springer, 2011), pp. 449–455
60. J.L. Thorogood, T.W. Hogg, H.S. Williamson, Application of risk analysis methods to subsurface well collisions. SPE Drill. Eng. **6**(4), 299–304 (1991). doi:10.2118/23941-PA
61. Y.H. Tsao, C.R. Dawson, D.W. Ure, Well collision avoidance, U.S. Patent 5 901 795, 11 May 1999
62. Z. Wang, T.A. Inglis, Planning directional wells through a high density cluster of existing wells. SPE Drill. Eng. **5**(4), 291–293 (1990). doi:10.2118/17594-PA

Chapter 14
Generating an Approximate Trivariate Spline Representation for Contractible Domains

Thien T. Nguyen

14.1 Introduction

A volume parameterization is a vector function that maps points in a three dimensional domain (physical domain) to points in a suitable domain (parameter domain), usually a cube, cylinder, sphere or PolyCube [21], a solid formed by joining several cubes face-to-face. Conversely, a parametric solid model is represented by a mapping from a parameter domain to the physical space. Therefore, once we know the parameterization mapping, we can construct the parametric model by approximating the inverse of the parameterization mapping. In order to build a valid (non-self-overlapping) model, the parametric mapping must be one-to-one. The parameterization mapping must therefore also be bijective on its image.

In this paper, we describe an algorithm to convert a contractible domain in $\mathbb{R}^3$, defined by its closed triangular mesh boundary, to a valid parametric trivariate B-spline model. The main goal is an application in isogeometric analysis, the new numerical scheme that provides the possibility of integrating finite element analysis into conventional NURBS-based CAD design tools [11]. So, our work is motivated by the fact that isogeometric analysis requires solid models represented by trivariate splines, while CAD systems usually only provide information on the boundary, i.e., mappings from the parametric domains to the boundary of the models [2].

T.T. Nguyen (✉)
University of Florida, Gainesville, FL, USA
e-mail: tuanthienbk@gmail.com

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_14

14.1.1 Related Work

Surface parameterization is mainly related to the problem of parameterizing triangulated surfaces. It has gained a lot of research interest in the past decades due to the advent of laser range scan technology which provided high-resolution models of physical objects with millions to billions of sample points. Basically, a good surface parameterization usually minimizes distortion in either angles or areas in some sense. An excellent review of many important methods as well as the fundamental theory is presented in the tutorial by Hormann et al. [10].

While many successes are achieved in the field of surface parameterization, there are only few works on volume parameterization and its theoretical guarantee for bijectivity is far from adequate. Several key limitations have been pointed out in [13] which prevent existing methods from being used in real applications. We mention here some of them which are related to the theoretical background of this paper. Martin et al. [17] proposed a parameterization method for a generalized cylinder-type volume defined by a tetrahedral mesh. This method is based on discrete harmonic functions and can be used for generating trivariate B-spline models for isogeometric analysis. Because the parameter domain is a cylinder, the parametrization is singular along the axis or the axis-like curve. The authors improved their algorithm in [16] to deal with this drawback. Aigner et al. [1] used least squares with several penalty terms corresponding to particular features of the shape to build a spline approximation for swept volumes. Such volumes include many free-form shapes in CAD system, like blades or propellers. Li et al. [13, 14] solved harmonic equations by the fundamental solution method to find a mapping between volumes with the same topology. This method is a bit slow and the results seem to depend on the fine-tuning of certain parameters and source points. In our method, we also use a meshless method to find a mapping, but based on the web-splines method [9]. Xia et al. [23] presented an idea to parameterize objects with handles to polycubes. In this method, two harmonic fields are first computed for both a 3-dimensional object and a corresponding polycube. Then by tracing integral curves generated by the harmonic fields, a parameterization is constructed. The authors proved the bijectivity of the mapping by its construction.

Volume parameterization is also an attractive approach for all-hex meshing where a volumetric mapping is exploited to transfer hex-mesh from a parameter domain to the input model. Gregson et al. present a method to compute PolyCube maps in [8]. CubeCover is another method for computing a volumetric map using a guiding frame-field [19]. Recently, Zhang and her co-authors have published a method to generate volumetric T-spline based on solving a harmonic field and PolyCube maps [15, 22].

14.1.2 Overview of Our Framework

Our framework consists of four steps.

- *Step 1 (Surface segmentation and parameterization).* The (triangular mesh) boundary $\partial\Omega$ is subdivided into six patches $\Gamma_i, i = 0, \ldots, 5$, with the connectivity of the facets of the unit cube. Then each patch is parameterized in order to obtain data for the next steps. Each two incident patches are C^0 matched at their sharing edge.
- *Step 2*. We find a mapping

$$\begin{aligned} f: \quad & \Omega \to [0,1]^3 \\ & \mathbf{p} \mapsto f(\mathbf{p}) = (f_1(\mathbf{p}), f_2(\mathbf{p}), f_3(\mathbf{p}))^T \end{aligned}$$

  by minimizing a sequence of harmonic mappings, i.e., we find f_1, then use f_1 to find f_2 and so on.
- *Step 3*. We modify the mapping f such that it is compatible to the surface parameterization found in Step 1. The modified mapping keeps all important properties from f, i.e., it is bijective.
- *Step 4*. Find the B-spline mapping $\mathbf{x} : [0,1]^3 \to \mathbb{R}$ such that $\mathbf{x} \approx f^{-1}$ by least-squares approximation.

Step 3 is the main new contribution in addition to the preliminary conference version [18]. In the next sections, we describe our framework step by step in detail.

14.2 Surface Segmentation and Parameterization

For a domain Ω defined as the interior of a closed triangular mesh, we would like to subdivide the boundary $\partial\Omega$ into six patches $\Gamma_i, i = 0, \ldots, 5$, as illustrated in Fig. 14.1. We use the method from [3]. In this method, users are allowed to choose some triangles as seeds for each patch. We use the $L^{2,1}$ metric as in the paper. Each triangle is associated to a cost to a particular seed. We construct a queue and use a region growing technique to obtain a segmentation result. The segmentation is topologically equivalent to the unit cube, i.e., each face has exactly four incident faces. After a segmentation of the boundary is generated, we parameterize each patch to obtain the mapping from the boundary of Ω to the boundary of the unit cube. We use the *mean value coordinate* method by Floater [7] for mesh parameterization. Then we may approximate each patch by its spline representation and consider these parametric surfaces as the boundary of Ω. In order to make C^0 continuity between each two adjacent patches, the control points on the shared edge are made coincident. Figure 14.1 shows six parametric patches in the spline approximation of the boundary mesh patches. We use the configuration of the boundary surfaces as illustrated in Fig. 14.2. For $i = 0, \ldots, 5$, we denote by

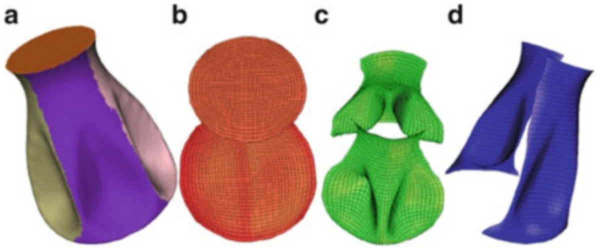


Fig. 14.1 Surface segmentation and parameterization. (**a**) Surface segmentation. (**b**) Γ_0, Γ_1. (**c**) Γ_2, Γ_3. (**d**) Γ_4, Γ_5

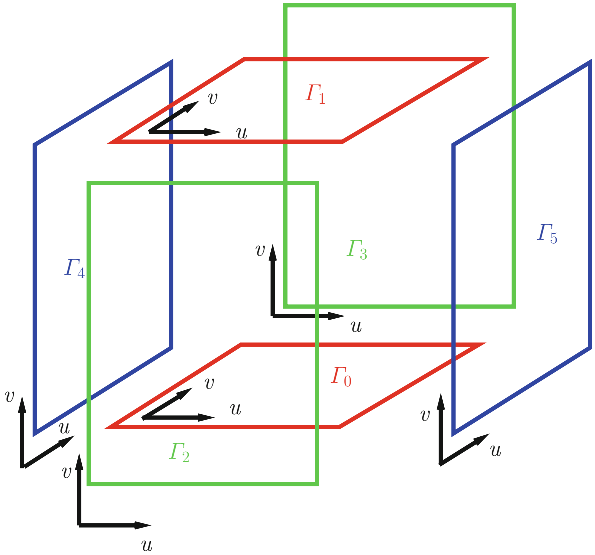


Fig. 14.2 Parameterization configuration of the boundary surfaces

$\gamma_i : \Gamma_i \to [0, 1]^2$ the parameterization mapping of the patch Γ_i and $\mathbf{x}_i : [0, 1]^2 \to \Gamma_i$ the parametric representation for each boundary surface, which is an approximation of γ_i^{-1}.

14.3 Mapping Defined by Sequences of Harmonic Maps

Once we obtain the suitable partition of the boundary into six patches and parameterization mappings, we proceed to Step 2. As described above, we are going to find the coordinate functions successively, i.e., we first find f_1, then use f_1 to find f_2 and finally find f_3 by using information from f_1 and f_2. In the following, we denote by $\Delta_{\mathcal{M}}$ and $\nabla_{\mathcal{M}}$ the Laplace-Beltrami operator and the tangential gradient operator on a particular manifold $\mathcal{M} \subset \mathbb{R}^3$.

14.3.1 The First Coordinate Function f_1

The function f_1 is designed as a critical point of the following harmonic energy

$$E_1(f_1) = \int_{\Omega} ||\nabla f_1||^2 \mathrm{dvol}^3(\Omega), \tag{14.1}$$

subject to the boundary conditions

$$f_1|_{\Gamma_0} = 0, \quad f_1|_{\Gamma_1} = 1$$

and

$$f_1 = (\gamma_i)_2 \text{ on } \Gamma_i, i = 2, 3, 4, 5.$$

Figure 14.3a illustrates the boundary conditions that f_1 must satisfy. So, the value of f_1 is 0 on the bottom surface, 1 on the top surface and increases gradually from 0 to 1 following the parametric curves on the remaining surface patches. Once f_1 is found, its level sets could be considered foliage layers piled up from the bottom to the top of the object, as shown in Fig. 14.3b. We denote by

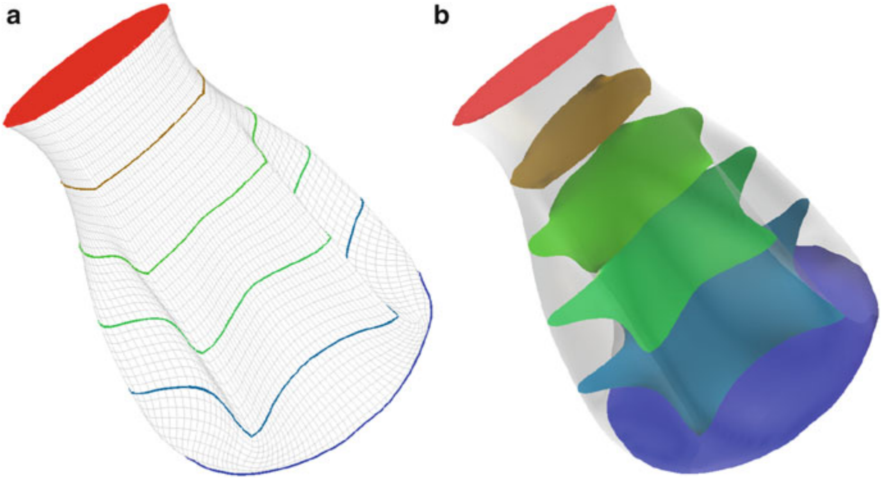


Fig. 14.3 The first coordinate function. (**a**) Boundary conditions. (**b**) The resulting level sets

$$L_1(z_1) = \{\mathbf{p} : f_1(\mathbf{p}) = z_1\}, z_1 \in [0,1],$$

the z_1-level set of f_1. Because f_1 is a harmonic map from Ω to $[0,1]$, we see that $L_1(z_1)$ is *contractible* for each $z_1 \in [0,1]$ by the maximum principle.

14.3.2 The Second Coordinate Function f_2

Once f_1 is found, we consider a single level set $L_1(z_1)$ for a particular $z_1 \in [0,1]$. We find f_2 as a critical point of the following harmonic energy

$$E_2(z_1, f_2) = \int_{L_1(z_1)} ||\nabla_{L_1(z_1)} f_2||^2 \mathrm{dvol}^2(L_1(z_1)), \tag{14.2}$$

subject to the boundary conditions

$$f_2|_{\Gamma_2 \cap L_1(z_1)} = 0, \;\; f_2|_{\Gamma_3 \cap L_1(z_1)} = 1$$

and

$$f_2 = (\gamma_i)_1 \text{ on } \Gamma_i \cap L_1(z_1), i = 4, 5.$$

Figure 14.4 illustrates the problem of f_2 in more detail. So, we see in this figure, the tangential gradient of f_2 on the level set $L_1(z_1)$ at a certain point is the projection of the gradient of f_2 onto the tangent plane at that point of $L_1(z_1)$. On $L_1(z_1)$, the value f_2 is 0 at the blue boundary curve, 1 at the red boundary curve and increases from 0 to 1 on the two remaining curves. Once the problem (14.2) is solved, an

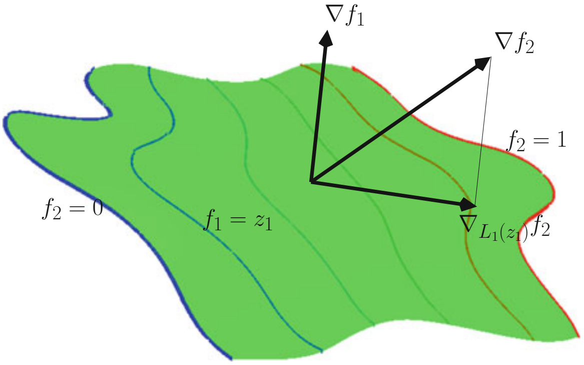


Fig. 14.4 The second coordinate function

intersection of a level set of f_2 and the surface $L(z_1)$ is a single arc, as depicted on Fig. 14.4. Indeed, if we denote by

$$L_{12}(z_1, z_2) = \{\mathbf{p} : f_1(\mathbf{p}) = z_1, f_2(\mathbf{p}) = z_2\}, (z_1, z_2) \in [0, 1]^2$$

the z_2-level set of f_2 on $L_1(z_1)$, then $L_{12}(z_1, z_2)$ is contractible for each $(z_1, z_2) \in [0, 1]^2$ by the maximum principle because f_2 is a harmonic map from the manifold $L_1(z_1)$ to $[0, 1]$.

14.3.3 The Third Coordinate Function f_3

Following the same strategy, we consider a single level set $L_{12}(z_1, z_2)$ for a particular $(z_1, z_2) \in [0, 1]^2$ and find f_3 as a critical point of the following harmonic energy

$$E_3(z_1, z_2, f_3) = \int_{L_{12}(z_1,z_2)} ||\nabla_{L_{12}(z_1,z_2)} f_3||^3 \mathrm{dvol}^1(L_{12}(z_1, z_2)) \tag{14.3}$$

subject to the boundary conditions

$$f_3|_{\Gamma_4 \cap L_{12}(z_1,z_2)} = 0, \ f_3|_{\Gamma_5 \cap L_{12}(z_1,z_2)} = 1.$$

Figure 14.5 illustrates the problem of f_3 in more detail. So, we see in this figure, the tangential gradient of f_3 on the level set $L_{12}(z_1, z_2)$ at a certain point is the projection of the gradient of f_3 onto the tangent line at that point of $L_{12}(z_1, z_2)$. A tangent vector at that point can be calculated by the cross product of the gradient of f_1 and the gradient of f_2, i.e. $\nabla f_1 \times \nabla f_2$. On $L_{12}(z_1, z_2)$, the value of f_3 is 0 at the blue point and 1 at the red point. Because f_3 is a harmonic mapping from the

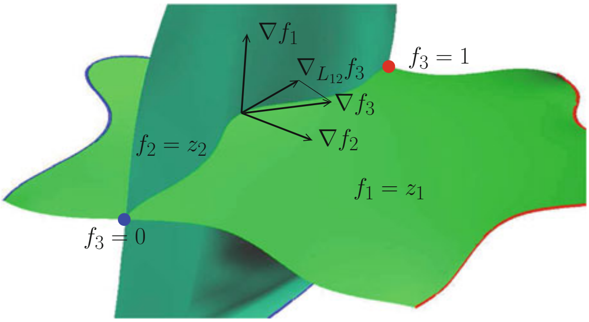


Fig. 14.5 The third coordinate function

manifold (in this case, it is a curve) $L_{12}(z_1, z_2)$ to $[0, 1]$, f_3 is an increasing function on this curve. Thus, if we denote by

$$L_{123}(z_1, z_2, z_3) = \{\mathbf{p} : f_1(\mathbf{p}) = z_1, f_2(\mathbf{p}) = z_2, f_3(\mathbf{p}) = z_3\}, (z_1, z_2, z_3) \in [0, 1]^3$$

the z_3-level set of f_3 in $L_{12}(z_1, z_2)$, then $L_{123}(z_1, z_2, z_3)$ is contractible for each $(z_1, z_2, z_3) \in [0, 1]^3$ by the maximum principle. Therefore $L_{123}(z_1, z_2, z_3)$ contains only a single point, i.e., $f = (f_1, f_2, f_3)^T$ is bijective. The following theorem summarizes the discussion above.

Theorem 14.1 *If f_1, f_2 and f_3 are the solutions of* (14.1)–(14.3)*, respectively, then the mapping $f = (f_1, f_2, f_3)^T$ from Ω to* $[0, 1]^3$ *is bijective.*

Now we turn our attention to the boundary conditions of (14.1)–(14.3). There is nothing special in (14.1) where the boundary conditions are applied to every point on the boundary of Ω. But in (14.2), if we can solve every problem for each $z_1 \in [0, 1]$, then the boundary conditions are applied only to four patches which are $\Gamma_2, \Gamma_3, \Gamma_4$ and Γ_5. There are no boundary conditions on two patches Γ_0 and Γ_1. In fact, we create a new parameterization for Γ_0 and Γ_1 because $L_1(0) = \Gamma_0$ and $L_1(1) = \Gamma_1$. If Γ_0 and Γ_1 have their own parameterizations γ_0 and γ_1, then $f_2 = (\gamma_i)_1, i = 1, 2$, if and only if $(\gamma_i)_1$ is also a critical point of the harmonic energy $E_2(z_1, f_2)$. This is due to the uniqueness property of harmonic mappings. Fortunately, because we use the mean value coordinate method to compute the parameterization of Γ_0 and Γ_1, the parameterizations satisfy Laplace's equations. Therefore, the function f_2 is compatible with the parameterization of the boundary of the domain Ω.

A similar observation happens in the problem (14.3). There are no boundary conditions on the four patches $\Gamma_0 = L_{12}(0, z_2)$, $\Gamma_1 = L_{12}(1, z_2)$, $\Gamma_2 = L_{12}(z_1, 0)$ and $\Gamma_3 = L_{12}(1, z_2)$. We create new parameterizations on these patches. Basically, f_3 is an arc-length parameterization scaled to $[0, 1]$ with respect to each curve $L_{12}(z_1, z_2), (z_1, z_2) \in [0, 1]^2$. The function $(\gamma_i)_2$ on each patch $\Gamma_i, i = 0, \ldots, 3$ is equal to f_3 if and only if $(\gamma_i)_2$ is also a critical point of the harmonic energy $E_3(z_1, z_2, f_3)$ on the iso-parametric curves in the v-direction of that patch where the parameterization function $(\gamma_i)_1$ must be identical to the coordinate function f_2.

To overcome the drawbacks as discussed above, we use a reparameterization technique to modify the function f_3 such that the modified function is compatible with the parameterization mapping on each boundary patch. We discuss in the next section more details of these modifications.

14.3.4 Numerical Implementation

To solve the problems (14.1)–(14.3), we use a method similar to the web-splines method [9]. More precisely, we find an approximate solution $f \approx \hat{f} = (\hat{f}_1, \hat{f}_2, \hat{f}_3)^T$ in the finite dimensional space spanned by B-splines. For finding an approximate solution $\hat{f}_l$, we write

$$\hat{f}_l = \sum_{(i,j,k)\in\boldsymbol{\mathcal{I}}} \underbrace{N_{i,\boldsymbol{\mathcal{X}}}(x)N_{j,\boldsymbol{\mathcal{Y}}}(y)N_{k,\boldsymbol{\mathcal{Z}}}(z)}_{B_{ijk}(x,y,z)}\, c^l_{i,j,k},\, l = 1,2,3$$

where $c^l_{i,j,k}$ are real coefficients for $l = 1,2,3$. The basis functions N_i, N_j and N_k are B-splines of degree d_1, d_2 and d_3 with respect to knot sequences $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{Y}}$ and $\boldsymbol{\mathcal{Z}}$ which are determined by the projections of the axis-aligned bounding box $\boldsymbol{\mathcal{B}}$ of Ω to the corresponding axis. In our implementation, we use uniform knot vectors in the x, y and z-direction and consider only those tensor-product basis functions whose supports overlap the domain. In order to determine the relevant basis functions, we compute a characteristic function χ of the domain Ω by an approximate implicitization algorithm, such as [12] or [5]. So, the boundary of the domain Ω is represented by the zero level set of an auxiliary function φ.

Now with the penalty method to enforce the boundary conditions, the problem (14.1) is written as

$$\lambda \int_{\partial\Omega} (\hat{f}_1 - f_{1\boldsymbol{\mathcal{D}}})^2 + \int_{\Omega} ||\nabla \hat{f}_1||^2 \to \min_{c^1_{i,j,k}}$$

where λ is a large weight, e.g., 10^3 and $f_{1\boldsymbol{\mathcal{D}}}$ is the function defined by Dirichlet boundary conditions of f on the boundary of Ω. The above minimization problem is quadratic with respect to c^1_{ijk}, so it leads to solving a linear system. We transform the integral over domain Ω to the integral over the whole space by using the computed characteristic function as follows

$$\int_{\Omega} ||\nabla \hat{f}_1||^2 = \int_{\mathbb{R}^3} \chi ||\nabla \hat{f}_1||^2$$

In order to avoid a linear system with high condition number when the supports of basis functions have only a small intersections with the domain, we set the value of χ to 1 on Ω where the value of φ is negative and to a small value, e.g. 10^{-3}, outside Ω where the value of φ is positive. We evaluate the integral by Gauss quadrature. The boundary integral term is evaluated by taking a weighted sum of finitely many points.

Once $\hat{f}_1$ is obtained, we proceed to solve the problem for f_2. It is much more difficult because f_2 is the critical point of harmonic energy on every level set of f_1. By means of minimizing simultaneously the harmonic energy for all level sets of f_1 contained in Ω, we integrate from 0 to 1 and use the co-area formula [6] to obtain

$$\int_0^1 \left(\int_{L_1(z_1)} ||\nabla_{L_1(z_1)} f_2||^2 \right) \mathrm{d}z_1 = \int_{\Omega} ||\nabla_{L_1(z_1)} f_2||^2 ||\nabla f_1||$$

The intrinsic gradient $\nabla_{L_1(z_1)} f_2$ is nothing else but the projection of the usual gradient onto the tangent space, see Fig. 14.4. We have

$$\nabla_{L_1(z_1)} f_2 = \mathbb{P}_{\nabla f_1} \nabla f_2 := \nabla f_2 - \frac{\nabla f_1 \cdot \nabla f_2}{\nabla f_1 \cdot \nabla f_1} \nabla f_1$$

Then by using the similar technique to the problem of finding $\hat{f}_1$, we arrive to the quadratic minimization problem with respect to the coefficients $c^2_{i,j,k}$. Solving a sparse linear system gives us the solution.

Once $\hat{f}_1$ and $\hat{f}_2$ are obtained, we proceed to solve the problem for f_3. Analogous to the problem of finding f_2, we minimize the harmonic energy of f_3 on every manifold which is the intersection of a level set of f_1 and a level set of f_2. This means that we integrate the harmonic energy on $[0, 1]^2$ and use the co-area formula. This yields

$$\int_{[0,1]^2} \left(\int_{L_{12}(z_1,z_2)} ||\nabla_{L_{12}(z_1,z_2)} f_3||^2 \right) \mathrm{d}z_1 \mathrm{d}z_2 = \int_{\Omega} ||\nabla_{L_{12}(z_1,z_2)} f_3||^2 ||\nabla f_1 \times \nabla f_2||$$

The intrinsic gradient $\nabla_{L_{12}(z_1,z_2)} f_3$, see Fig. 14.5, is calculated by

$$\nabla_{L_{12}(z_1,z_2)} f_3 = \mathbb{P}_{\nabla f_1 \times \nabla f_2} \nabla f_3 := (\nabla f_3 \cdot \mathbf{t})\mathbf{t}$$

where the tangential vector is computed by

$$\mathbf{t} = \frac{\nabla f_1 \times \nabla f_2}{||\nabla f_1 \times \nabla f_2||}.$$

Finally, similar to the problem of finding $\hat{f}_1$ and $\hat{f}_2$, we have to solve a quadratic minimization problem to obtain the coefficients $c^3_{i,j,k}$.

14.4 Boundary Parameterizations Preserving by Reparameterization

First, we explain what is a reparameterization mapping for a parametric surface by the following definition:

Definition 14.1 Let $\mathcal{S} := \{s(u, v) \in \mathbb{R}^3 : (u, v) \in \mathcal{D} \subset \mathbb{R}^2\}$ be an arbitrary parametric surface. The mapping $R : \hat{\mathcal{D}} \to \mathcal{D}, \hat{\mathcal{D}} \subset \mathbb{R}^2$ is called a reparameterization mapping if it satisfies

- R is bijective.
- R maps the boundary of $\hat{\mathcal{D}}$ to the boundary of $\mathcal{D}$ or $R(\partial\hat{\mathcal{D}}) = \partial\mathcal{D}$.

After we obtain two coordinate functions f_1 and f_2, we see that the parameterizations of Γ_4 and Γ_5 are preserved. Moreover, the function f_1 also reproduces the $(\gamma_i)_2$ of Γ_i with $i = 2, 3$, and the function f_2 reproduces the $(\gamma_i)_2$ of Γ_i with $i = 0, 1$. Then we have to modify f_3 by reparameterization functions in order to preserve the parameterizations of $\Gamma_i, i = 0, \ldots, 3$. For instance, we know that (f_3, f_2) produces a parameterization for Γ_0 and we want to have a reparameterization $R : [0, 1]^2 \to [0, 1]^2$ such that $R(f_3, f_2)$ matches the original parameterization γ_0 of the boundary patch Γ_0. More precisely, if $\mathbf{p} \in \Gamma_0$, we have $R(f_3(\mathbf{p}), f_2(\mathbf{p})) = (r_0(f_3(\mathbf{p}), f_2(\mathbf{p})), f_2(\mathbf{p}))$. Here the second coordinate of R is f_2 because f_2 reproduces the $(\gamma_0)_2$ on Γ_0, and we only have to compute the reparameterization function r_0.

14.4.1 Designing Reparameterization Functions

We prove that the mapping f with modified f_3 is still bijective. In the proof, we use another criterion of the bijectivity of a smooth mapping which is stated in the following theorem

Theorem 14.2 ([24]) *Suppose that f is a C^1 volumetric mapping from a compact domain $\Omega \subset \mathbb{R}^n$ with a connected boundary to a topologically equivalent parameter domain in $\mathbb{R}^n$. If f is bijective on the boundary $\partial\Omega$ and its Jacobian $J_f = \det|\nabla f|$ does not vanish on Ω, then f is bijective.*

The next theorem describes our reparameterization technique.

Theorem 14.3 *Suppose that $f = (f_1, f_2, f_3)^T$ solves the Eqs.* (14.1)–(14.3) *and f_3 is modified in two steps as follows*

$$f_3^{+} = r_0(f_3, f_2)(1 - f_1) + r_1(f_3, f_2) f_1 \tag{14.4}$$

$$f_3^{++} = r_2(f_3^{+}, f_1)(1 - f_2) + r_3(f_3^{+}, f_1) f_2 \tag{14.5}$$

where $r_i : [0, 1]^2 \to [0, 1]$ are the reparameterization functions on $\Gamma_i, i = 0, \ldots, 3$. Then the modified mapping $f = (f_1, f_2, f_3^{++})^T$ is bijective and preserves the boundary parameterizations.

Proof It is easy to see that the mapping $f = (f_1, f_2, f_3)^T$ is bijective by the same arguments as in the previous section. Without loss of generality, we assume that the Jacobian of f is positive. Now we prove that (f_1, f_2, f_3^{++}) actually preserves the parameterizations of boundary surfaces.

First, it is observed that $f_3^{+} = r_0(f_3, f_2)$ on Γ_0 where $f_1 = 0$ and $f_3^{+} = r_1(f_3, f_2)$ on Γ_1 where $f_1 = 1$. Also, $f_3^{++} = r_2(f_3^{+}, f_2)$ on Γ_2 where $f_2 = 0$ and $f_3^{++} = r_3(f_3^{+}, f_2)$ on Γ_3 where $f_2 = 1$. Thus, f_3^{+} is compatible with the parameterizations of Γ_0 and Γ_1 and f_3^{++} is compatible with the parameterizations of Γ_2 and Γ_3. We show $f_3^{++} = f_3^{+}$ on Γ_0 and Γ_1. Considering $\mathbf{p} \in \Gamma_0 \cap \Gamma_2$, we have $f_1(\mathbf{p}) = 0$, $f_3^+(\mathbf{p}) = (\gamma_2)_2(\mathbf{p})$ and $r_2(f_3^+(\mathbf{p}), f_1(\mathbf{p})) = (\gamma_2)_2(\mathbf{p})$. This means that $r_2(u, 0) = u, \forall u \in [0, 1]$. Analogously, we have $r_2(u, 1) = u$, $r_3(u, 0) = u$ and $r_3(u, 1) = u$ for all $u \in [0, 1]$. Then, on Γ_0 i.e. $f_1 = 0$, we have $r_2(f_3^+, f_1) = f_3^+$ and $r_3(f_3^+, f_1) = f_3^+$. Therefore, by (14.5) $f_3^{++} = f_3^+$ on Γ_0. Similarly, $f_3^{++} = f_3^+$ on Γ_1.

It remains to be shown that $(f_1, f_2, f_3^{++})^T$ is bijective. That is, we will show the Jacobian of $(f_1, f_2, f_3^{++})^T$ calculated by the triple product

$$[\nabla f_1, \nabla f_2, \nabla f_3^{++}] = (\nabla f_1 \times \nabla f_2) \cdot \nabla f_3^{++}$$

is larger than zero. To do this, we calculate the gradient of f_3^+ by

$$\begin{aligned}\nabla f_3^+ =& \nabla f_3\Big(\partial_u r_0(f_3, f_2)(1 - f_1) + \partial_u r_1(f_3, f_2) f_1\Big)\\ &+ \nabla f_2\Big(\partial_v r_0(f_3, f_2)(1 - f_1) + \partial_v r_1(f_3, f_2) f_1\Big)\\ &+ \nabla f_1(r_1(f_3, f_2) - r_0(f_3, f_2))\end{aligned}$$

Then we have

$$\begin{aligned}[\nabla f_1, \nabla f_2, &\nabla f_3^+] = [\nabla f_1, \nabla f_2, \nabla f_3].\\ &\cdot\Big(\partial_u r_0(f_2, f_3)(1 - f_1) + \partial_u r_1(f_2, f_3) f_1\Big)\end{aligned} \tag{14.6}$$

On Γ_0, we know $f_1 = 0$ and $[\nabla f_1, \nabla f_2, \nabla f_3^+] > 0$, because $(f_1, f_2, f_3^+)^T$ produces exactly the parameterization of Γ_0 and that parameterization is regular. One deduces from (14.6) with $J_f = [\nabla f_1, \nabla f_2, \nabla f_3] > 0$ that $\partial_u r_0(u, v) > 0, \forall(u, v) \in [0, 1]^2$. By similar arguments on Γ_1, we have $\partial_u r_1(u, v) > 0, \forall(u, v) \in [0, 1]^2$. By (14.6), this yields

$$[\nabla f_1, \nabla f_2, \nabla f_3^+] > 0, \text{ on } \Omega.$$

Now, we calculate the gradient of f_3^{++} and obtain, using similar arguments, the following:

$$[\nabla f_1, \nabla f_2, \nabla f_3^{++}] = [\nabla f_1, \nabla f_2, \nabla f_3^+] \underbrace{[\cdots]}_{>0} > 0, \text{ on } \Omega.$$

14.4.2 Numerical Implementation

We describe here how to compute the reparameterization functions $r_0(u, v), r_1(u, v)$ in (14.4) and $r_2(u, v), r_3(u, v)$ in (14.5). We use the well-known least squares fitting methods. First, the reparameterization function r_i is represented as a linear combination of basis functions defined on $[0, 1]^2$, i.e.

$$r_i(u, v) = \sum_{j=1}^{n_i} \alpha_j^i B_j^i(u, v), \quad i = 0, \dots, 3$$

Next, we consider the problem of finding r_0. By its definition, if we have a point $\mathbf{p} \in \Gamma_0$, we wish $r_0(f_3(\mathbf{p}), f_2(\mathbf{p})) = (\gamma_0)_1(\mathbf{p})$. For this reason, we formulate the following least squares problem

$$\sum_{k=1}^{n} \Big(r_0(f_3(\mathbf{p}_k), f_2(\mathbf{p}_k)) - (\gamma_0)_1(\mathbf{p}_k) \Big)^2 + \textit{ regularization terms } \rightarrow \min_{\alpha_j^0}$$

where the *regularization terms* can be defined as

$$w \int_0^1 \int_0^1 [(r_0)_{uu}^2 + 2(r_0)_{uv}^2 + (r_0)_{vv}^2] \mathrm{d}u \mathrm{d}v$$

with a small weight w. The least squares problem leads to solving a linear system to find a solution. The reparameterization function r_1 is found in the similar way. Once r_0 and r_1 are found, we can compute f_3^+ by the formula (14.4). Then, we apply again the least squares fitting techniques to find the next reparameterization function r_2. More precisely, we solve the following minimization problem

$$\sum_{k=1}^{n} \Big(r_2(f_3^+(\mathbf{p}_k), f_1(\mathbf{p}_k)) - (\gamma_2)_1(\mathbf{p}_k) \Big)^2 + \textit{ regularization terms } \rightarrow \min_{\alpha_j^2}$$

with the sample points $\mathbf{p}_k \in \Gamma_2$. Solving a linear system gives us a solution. Finally, the last reparameterization function r_3 is found by the same techniques.

14.5 Spline Approximation

This is the last step in our framework, once we obtain a boundary-parameterization-preserving mapping. So, we are going to find the mapping $\mathbf{x} \approx f^{-1}$ in the tensor product B-spline form as

$$\begin{aligned} \mathbf{x}(u, v, w) &= \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \sum_{k=0}^{n_3} N_{i,p}^{\mathcal{U}}(u) N_{j,q}^{\mathcal{V}}(v) N_{k,r}^{\mathcal{W}}(w) \mathbf{d}_{i,j,k} \\ &= B(u, v, w) \cdot D, \ (u, v, w) \in [0, 1]^3, \end{aligned}$$

where $D = (\mathbf{d}_{i,j,k})_{ijk}$ is the vector of control points and $B(u,v,w) = (B_{i,j,k}(u,v,w))_{ijk}$ is the vector of basis functions. Here we denote by $B_{i,j,k}(u,v,w) = N^{\mathcal{U}}_{i,p}(u)N^{\mathcal{V}}_{j,q}(v)N^{\mathcal{W}}_{k,r}(w)$ the tensor product B-splines of degree p,q and r defined on three given knot vectors $\mathcal{U},\mathcal{V}$ and $\mathcal{W}$. Here, we have two options. We can either use arbitrary basis functions as we want (usually defined on uniform knot vector) or approximate boundary surface patches first by bivariate B-splines then keep the boundary control points fixed during volume fitting process. In the later case, the knot vectors of boundary surfaces should be compatible with each other, if not we can use the knot insertion algorithm to make them compatible. Now we formulate the problem of finding $\mathbf{x}$ as the following least squares problem

$$\int_{\Omega} ||(\mathbf{x} \circ f)(\mathbf{p}) - \mathbf{p}||^2 \mathrm{d}\mathbf{p} + R_{\mathbf{x}} \to \min_{D} \tag{14.7}$$

where $R_{\mathbf{x}}$ is the fairness term

$$R_{\mathbf{x}} = \omega_1 \int_{[0,1]^3} (\mathbf{x}_u^2 + \mathbf{x}_v^2 + \mathbf{x}_w^2)\mathrm{d}u\mathrm{d}v\mathrm{d}w + \omega_2 \int_{[0,1]^3} (\mathbf{x}_{uu}^2 + \\ \mathbf{x}_{vv}^2 + \mathbf{x}_{ww}^2 + 2\mathbf{x}_{uv}^2 + 2\mathbf{x}_{vw}^2 + 2\mathbf{x}_{wu}^2)\mathrm{d}u\mathrm{d}v\mathrm{d}w$$

and ω_1 and ω_2 are weights. For the later case described above, the boundary control points are known and are identical to the control points of the boundary surfaces. So, we only have to compute the inner control points. The characteristic function χ for Ω is used again to transform the integral over Ω to the integral over $\mathbb{R}^3$. We already computed χ in the approximate implicitization step but now it is set to 0 outside Ω. We have

$$\int_{\Omega} ||(\mathbf{x} \circ f)(\mathbf{p}) - \mathbf{p}||^2 \mathrm{d}\mathbf{p} = \int_{\mathbb{R}^3} ||(\mathbf{x} \circ f)(\mathbf{p}) - \mathbf{p}||^2 \chi(\mathbf{p})\mathrm{d}\mathbf{p}$$

We evaluate the integral by numerical quadrature, taking uniform grid of points on the bounding box $\mathcal{B}$ of Ω.

The least squares problem is a quadratic minimization problem with respect to the unknowns (which are the inner control points for the latter case). Therefore, it is relatively simple to find the solution by solving a sparse linear system.

14.6 Results

We summarize the implementation of our framework step by step. Then we show results from our method applied to several models, including a screw driver, a molecule and a femur.

14.6.1 Putting Things Together

First, a contractible domain defined by its closed triangular mesh boundary is given. Next, we have to do segmentation and parameterization to make the boundary $\partial\Omega$ topology equivalent to $[0, 1]^3$. Then, we find the bounding box of Ω. It is the domain for the web-splines method for computing the parameterization mapping. Next, we generate the characteristic function for Ω by an approximate implicitization method. Finally, the three variational problems for coordinate functions, reparameterization and spline approximation as described above, are performed.

14.6.2 Examples

Our method is implemented in *C*++ and uses *Gotools* [20] for B-splines manipulations. In addition, for solving the various arising sparse linear systems, we employ the *umfpack* routine [4]. The results in this paper are visualized using *Coin 3D* and *Qt*.

Example 1 The first model we use to test our method is the base of a screw driver given in triangular mesh. Figure 14.6 summarizes our method step by step. In this example, we approximate six boundary patches by uniform tensor-product quadratic B-splines with 32×32 control points. So, the resulting trivariate splines are uniform and quadratic with $32 \times 32 \times 32$ control points. We use the same configuration in the next two examples. In Fig. 14.6d, we see that the discrete Coons patch method gives us a solution with self-overlapping. If we do the boundary parameterization preserving step before finding $\mathbf{x}$, we will get a result with preserving boundary like Fig. 14.6f, otherwise we will get the object with a jagged boundary like Fig. 14.6e. We observe that both results in (e) and (f) have non-self-overlapping grids inside the object.

Example 2 Figure 14.7 shows the results from our method applied to a molecule model. The boundary of this model is smooth. Therefore, vanishing Jacobians are expected to happen around the common edges of boundary surfaces. Like the previous example, boundary parameterization preserving give us a better result.

Example 3 The last example we want to show is the femur model as shown in Fig. 14.8. This model is rather more complicated than those of the previous two examples. It seems that if we work directly on the mesh model, i.e. the former case in the spline approximation section, instead of using approximate bivariate spline boundary patches, we will get a result with better approximation for the true model. However, the grid inside seems to be more jagged.

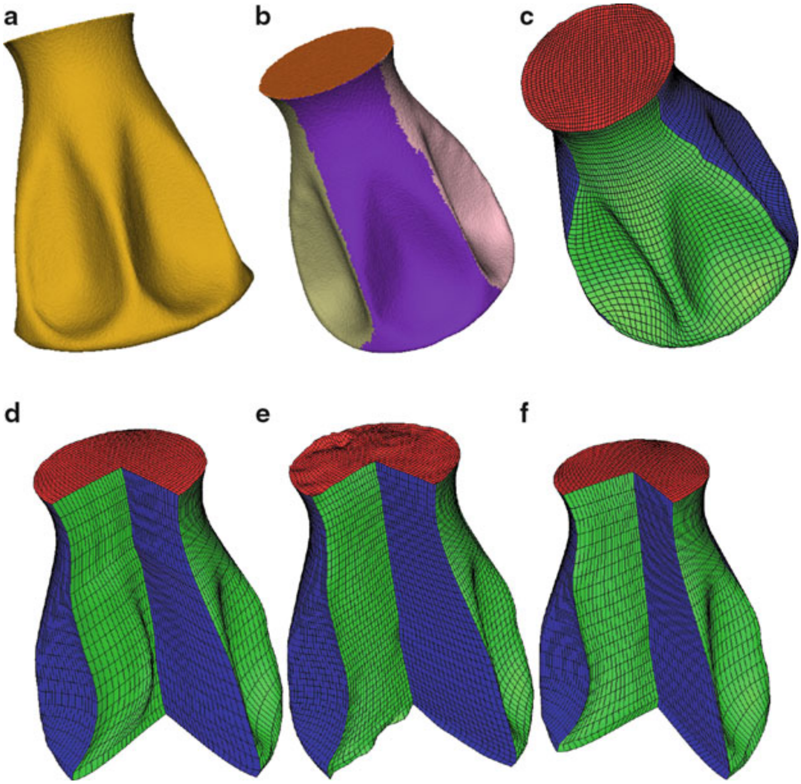


Fig. 14.6 The base of a screw driver model. (**a**) Triangular mesh. (**b**) Mesh segmentation. (**c**) Surface parameterization and approximation. (**d**) Discrete Coons patch. (**e**) Non boundary conforming. (**f**) Boundary conforming

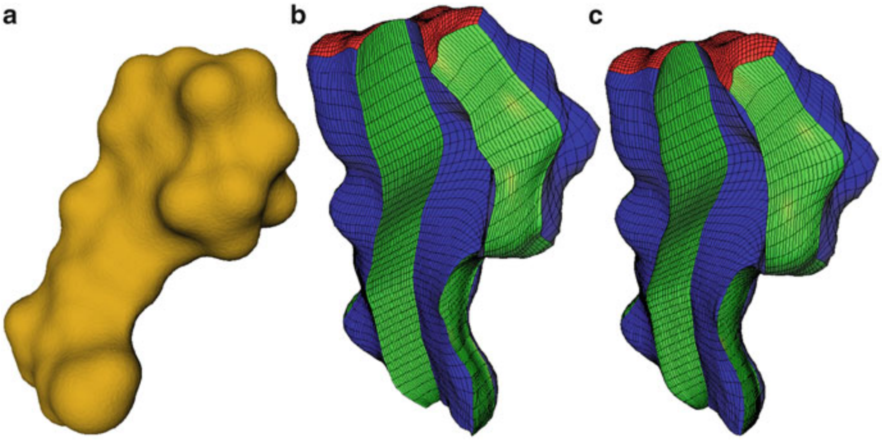


Fig. 14.7 The molecule model. (**a**) Triangular mesh. (**b**) Non boundary conforming. (**c**) Boundary conforming

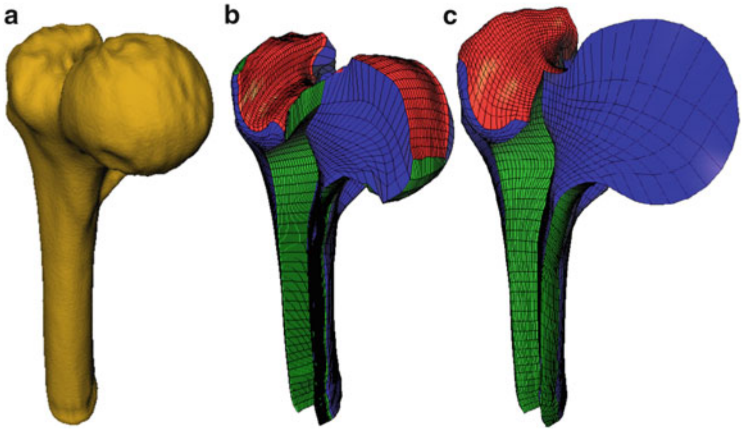


Fig. 14.8 The femur model. (**a**) Triangular mesh. (**b**) Directly on the mesh. (**c**) Boundary conforming

Conclusion

In this paper, we proposed a framework to compute a bijective mapping from a domain defined by its boundary to the unit cube in $\mathbb{R}^3$. This mapping is defined via a sequence of harmonic maps and modified by a reparameterization to conform with the boundary parameterization. The final spline representation for the domain was constructed as an approximation of the inverse of the computed mapping. We also demonstrated that our method works efficiently for some complicated domains.

In the future, we wish to improve the feature-aligned boundary segmentation. We also want to pull back the framework to a parameter space so that the numerical computation could be performed more efficient. Last but not least, we would like to develop a method for objects with handles.

References

1. M. Aigner, C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon, A.V. Vuong, Swept volume parameterization for isogeometric analysis, in *Proceedings of the 13th IMA International Conference on Mathematics of Surfaces XIII* (Springer, Cambridge, UK, 2009), pp 19–44
2. E. Cohen, T. Martin, R.M. Martin, T.Lyche, R.F. Riesenfeld, Analysis-aware modeling: understanding quality considerations in modeling for isogeometric analysis. Comput. Methods Appl. Mech. Eng. **199**(5), 334–356, (2010)
3. D. Cohen-Steiner, P. Alliez, M. Desbrun, Variational shape approximation. ACM Trans. Graph. **23**, 905–914 (2004)

4. T. Davis, UMFPACK: unsymmetric multifrontal sparse LU factorization package. http://www.cise.ufl.edu/research/sparse/umfpack/
5. T. Dokken, J.B. Thomassen, Weak approximate implicitization, in *Shape Modeling and Applications, International Conference on Shape Modeling and Applications (SMI'06, Matsushima, Japan)*, 0:31, 2006
6. H. Federer, Curvature measures. Trans. Am. Math. Soc. **93**, 418–491 (1959)
7. M.S. Floater, Mean value coordinates. Comput. Aided Geom. Des. **20**(1), 19–27 (2003)
8. J. Gregson, A. Sheffer, E. Zhang, All-Hex mesh generation via volumetric PolyCube deformation. Comput. Graph. Forum **30**, 1407–1416 (2011). doi:10.1111/j.1467-8659.2011.02015.x
9. K. Höllig, *Finite Element Methods with B-splines*. Frontiers in Applied Mathematics, vol. 26 (Society for Industrial and Applied Mathematics, Philadelphia, 2003)
10. K. Hormann, K. Polthier, A. Sheffer, Mesh parameterization: theory and practice, in *SIGGRAPH Asia 2008 Course Notes*, Singapore, no. 11, Dec 2008 (ACM, 2008)
11. T.J.R Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. Comput. Methods Appl. Mech. Eng. **194**(39–41), 4135–4195 (2005)
12. B. Jüttler, A. Felis, Least-squares fitting of algebraic surfaces. Adv. Comput. Math. **17**, 135–152 (2002)
13. X. Li, X. Guo, H. Wang, Y. He, X. Gu, H. Qin, Harmonic volumetric mapping for solid modeling applications, in *Shape Modeling International Conference (SMI, Lyon, France)*, 2007, pp. 109–120
14. X. Li, H. Xu, S. Wan, Z. Yin, W. Yu, Feature-aligned harmonic volumetric mapping using MFS. Comput. Graph. **34**(3), 242–251 (2010)
15. L. Liu, Y. Zhang, T.J.R. Hughes, M.A. Scott, T.W. Sederberg, Volumetric T-spline construction using boolean operations. Eng. Comput. (2013). doi:10.1007/s00366-013-0346-6
16. T. Martin, E. Cohen, Volumetric parameterization of complex objects by respecting multiple materials. Comput. Graph. **34**(3), 187–197 (2010)
17. T. Martin, E. Cohen, R.M. Kirby, Volumetric parameterization and trivariate B-spline fitting using harmonic functions. Comput. Aided Geom. Des. **26**(6), 648–664 (2009)
18. T. Nguyen, B. Jüttler, Parameterization of contractible domains using sequences of harmonic maps. in *Proceedings of Curves and Surfaces* (Springer, Avignon, France, 2010)
19. M. Nieser, U. Reitebuch, K. Polthier, CubeCover-parameterization of 3D volumes. Comput. Graph. Forum **30**, 1397–1406 (2011). doi:10.1111/j.1467-8659.2011.02014.x
20. SINTEF, GoTools, http://www.sintef.no/Projectweb/Geometry-Toolkits/GoTools/
21. M. Tarini, K. Hormann, P. Cignoni, C. Montani, PolyCube-maps, in *Proceedings of SIGGRAPH*, (Los Angeles, California, US, 2004), pp. 853–860, 1, 3
22. W. Wang, Y. Zhang, L. Liu, T.J.R. Hughes, Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. Comput. Aided Des. **45**(2), 351–360 (2013). A Special Issue of Solid and Physical Modeling (2012)
23. J. Xia, Y. He, X. Yin, S. Han, X. Gu, Direct-product volumetric parameterization of handlebodies via harmonic fields, in *Shape Modeling International Conference (SMI, Aix-En Provence, France)*, June 2010, pp. 3–12
24. G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Parameterization of computational domain in isogeometric analysis: methods and comparison. Comput. Methods Appl. Mech. Eng. **200**(23–24), 2021–2031 (2011)

Chapter 15
Isogeometric Analysis of Navier-Stokes Flow Using Locally Refinable B-Splines

Peter Nørtoft and Tor Dokken

15.1 Introduction

In recent years, isogeometric analysis (IGA) has gained increasing interest as a numerical method for solving engineering problems within fluid mechanics [7, 15]. This popularity may be attributed to its ability to model complex geometries exactly, to approximate the flow fields with arbitrarily high degree of smoothness, and to couple the geometric modeling and the flow analysis into one single framework. At the very heart of the isogeometric paradigm is the unification of finite element analysis (FEA) for solving the governing flow equations, and computer-aided design (CAD) for modeling the geometry of the flow domain.

One of the early challenges of the isogeometric paradigm was the concept of *local refinement*. To resolve the flow around some obstacle, say, a *fine* approximation of the field is often required in the boundary layer close to the obstacle, whereas a *coarse* approximation suffices in the far-field away from the obstacle. Here, efficient local refinement is of paramount importance: a coarse representation of the entire flow domain leaves the boundary layer unresolved, and the results are useless; a fine representation of the entire flow domain yields a fatal blow-up in the number of degrees of freedom, rendering the approach useless. This is sketched in Fig. 15.1. Although well-established within FEA, efficient local refinement was initially prohibited in IGA, primarily owing to the tensor-product structures inherited from CAD, that only allowed for global refinements, or a "poor man's" local refinement

P. Nørtoft (✉)
DTU Compute, Technical University of Denmark, Matematiktorvet 303B, DK-2800 Kgs. Lyngby, Denmark
e-mail: penn@dtu.dk

T. Dokken
SINTEF ICT, P.O. Box 124 Blindern, N-0314 Oslo, Norway
e-mail: Tor.Dokken@sintef.no

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_15

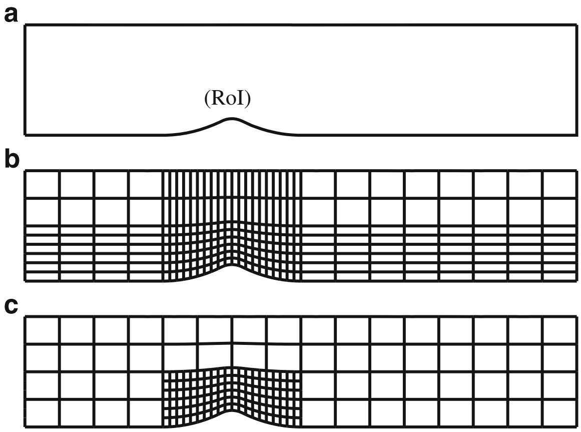


Fig. 15.1 Difference between global (**b**) and local (**c**) refinement of a region of interest (*RoI*) inside a flow domain (**a**)

through patching. Several ways to achieve local refinement in an isogeometric setting have been proposed since the birth of IGA, including in particular T-splines [2, 9, 18], and hierarchical splines [14, 19].

In this work, we study a novel approach to local refinement in the context of fluid mechanics, namely through the recently proposed locally refinable (LR) B-splines [4, 8, 16]. We investigate two families of locally refined B-spline discretizations of the flow variables for solving the mixed formulation of the stationary, incompressible Navier-Stokes equations in 2 dimensions using IGA. These two LR B-spline discretizations are motivated by recent results for ordinary tensor-product B-spline discretizations of the flow variables [3, 5, 17]. Our focus here is primarily on *how* to refine the flow discretizations, as dictated by a refinement strategy, and not on *whether* or *where* to refine, as dictated by some error estimator. In the context of a full-blown adaptive mesh refinement setup, of course, one needs to address all of these issues.

The outline of the rest of the work is as follows. We start by introducing the governing Navier-Stokes equation in Sect. 15.2, after which we introduce the LR B-splines as refinable building blocks for solving it in Sect. 15.3. Then, in Sect. 15.4, we briefly outline the general isogeometric framework, and in Sect. 15.5 we present the two flow discretizations. Numerical examples are presented in Sect. 15.6 with focus on the numerical stability, error convergence during refinements, and benchmarking. Finally, we summarize our findings and outline future work.

15.2 Navier-Stokes Equation

We start by introducing the steady-state, incompressible Navier-Stokes equation. This is the equation that governs the motion of fluids under sufficiently simple conditions.

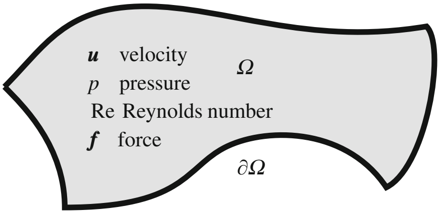


Fig. 15.2 A fluid contained in a flow domain

We consider a fluid in a 2-dimensional domain Ω as depicted in Fig. 15.2. We assume the fluid is isothermal, i.e., at constant temperature, incompressible, i.e., the density is constant, and Newtonian, i.e., the stress and the strain rate are linearly related through the viscosity, which is also assumed to be constant. Finally, we assume that the flow is stationary, i.e., time-independent. The state of the fluid is then given by the velocity and the pressure, and these are governed by the Navier-Stokes and mass-continuity equations:

$$(\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} + \nabla p - \frac{1}{\mathrm{Re}} \Delta \boldsymbol{u} + \boldsymbol{f} = \boldsymbol{0}, \tag{15.1a}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{15.1b}$$

Here $\boldsymbol{u} = (u, v)$ is the velocity, p is the pressure, and $\boldsymbol{f}$ are additional body forces acting on the fluid, all in dimensionless form, while $\mathrm{Re} := \rho U L / \mu$ is the Reynolds number, where ρ is the density, μ is the viscosity, and U and L are characteristic velocity and length scales of the problem, respectively. In somewhat loose terms, Re is a measure of the degree of nonlinearity, and hence complexity, of the flow problem. We will consider primarily *laminar* flows with $\mathrm{Re} \lesssim 2{,}000$, as opposed to *turbulent* flows with $\mathrm{Re} \gtrsim 2{,}000$.

These equations govern the flow in the interior of the domain Ω, and they must be augmented by suitable boundary conditions. Here, we consider so-called full Dirichlet boundary conditions, along with a condition on the average pressure:

$$\boldsymbol{u} = \boldsymbol{g} \qquad \text{on } \partial\Omega, \tag{15.2a}$$

$$\int_{\Omega} p \, \mathrm{d}\boldsymbol{x} = p_0. \tag{15.2b}$$

15.3 Locally Refinable B-Splines

We now proceed to give a brief introduction to Locally Refinable (LR) B-splines. First, the underlying B-splines are introduced, after which their LR extensions are described in 2 dimensions. The intention is to give the reader an intuitive understanding of LR B-splines, and thereby pave the road for using them to solve the Navier-Stokes equation in the next sections. For a more rigorous introduction to LR B-splines, we refer to [8].

15.3.1 B-Splines

We start by recalling the definition of univariate B-splines. Given a polynomial degree $d \geq 0$ and a non-decreasing sequence of $d + 2$ knots $\boldsymbol{\eta} = \{\eta_1, \dots, \eta_{d+2}\}$, a univariate B-spline $B[\boldsymbol{\eta}] : \mathbb{R} \to \mathbb{R}$ is defined recursively through:

$$B[\boldsymbol{\eta}](\xi) = \frac{\xi - \eta_1}{\eta_{d+1} - \eta_1} B[\eta_1, \dots, \eta_{d+1}](\xi) + \frac{\eta_{d+2} - \xi}{\eta_{d+2} - \eta_2} B[\eta_2, \dots, \eta_{d+2}](\xi), \tag{15.3a}$$

starting with

$$B[\eta_i, \eta_{i+1}](\xi) = \begin{cases} 1 & \text{if } \eta_i \leq \xi < \eta_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{15.3b}$$

for $i = 1, \dots, d + 1$, and where terms with zero denominator are defined to be zero. A univariate B-spline is thus a piecewise polynomial function of degree d. Its support is the interval $[\eta_1, \eta_{d+2}]$, and the continuity across a knot η_i is $d - m$, where m denotes the multiplicity of the knot η_i.

Multivariate B-splines can be formed quite naturally through tensor-product structures based on multiple univariate B-splines. In two parametric dimensions we have the following: Given two polynomial degrees d_i and two non-decreasing sequences of $d_i + 2$ knots $\boldsymbol{\eta}_i = \{\eta_{i,1}, \dots, \eta_{i,d_i+2}\}$ for $i = 1, 2$, a bivariate tensor-product B-spline $B[\boldsymbol{\eta}_1, \boldsymbol{\eta}_2] : \mathbb{R}^2 \to \mathbb{R}$ is given by:

$$B[\boldsymbol{\eta}_1, \boldsymbol{\eta}_2](\xi_1, \xi_2) = B[\boldsymbol{\eta}_1](\xi_1)\, B[\boldsymbol{\eta}_2](\xi_2). \tag{15.4}$$

The support of a tensor-product B-spline is the rectangle $[\eta_{1,1}, \eta_{1,d_1+2}] \times [\eta_{2,1}, \eta_{2,d_2+2}]$. Figure 15.3 illustrates the construction of a bivariate tensor-product B-spline from two univariate B-splines. In the example shown, the two univariate B-splines that form the bivariate bi-quadratic tensor-product B-spline are constructed from the polynomial degrees $d_1 = d_2 = 2$, and the knot vectors $\boldsymbol{\eta}_1 = \{0, 1/3, 2/3, 1\}$ and $\boldsymbol{\eta}_2 = \{0, 0, 1/3, 2/3\}$. The knot vectors $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ are extracted from two identical global knot vectors $\overline{\boldsymbol{\eta}}_1 = \overline{\boldsymbol{\eta}}_2 = \{0, 0, 0, 1/3, 2/3, 1, 1, 1\}$. Each of these gives rise to 5 univariate B-splines, resulting in a total of 25 bivariate B-splines.

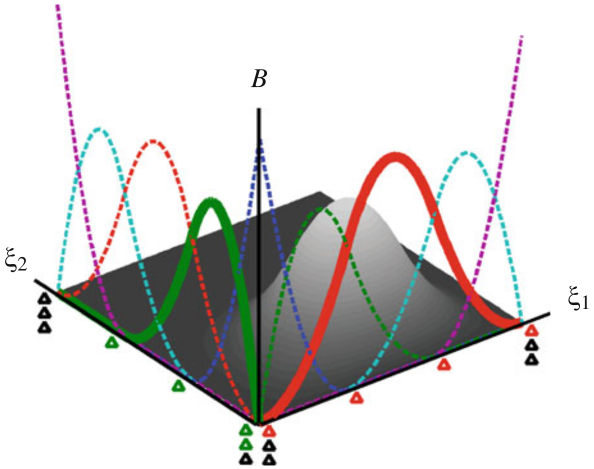


Fig. 15.3 Construction of a bi-quadratic tensor-product B-spline (*grayscale surface*) from two univariate quadratic B-splines (*lines in bold*) with given knot vectors (*colored triangles*)

15.3.2 LR Mesh

Locally refinable B-splines rest naturally on B-splines. They include the tensor-product B-splines introduced above as a special case, but in addition provide a much more "local" framework for multivariate B-splines. Before understanding the notion of an LR B-spline, we must, however, understand the notion of an LR *mesh*.

Just like any tensor-product B-spline is formed on a tensor-product mesh, as sketched in Fig. 15.3, any LR B-spline is formed on an LR mesh. A mesh holds information about essentially two things: the *location* and the *multiplicity* of all knots. For tensor-product B-splines, as defined in (15.4), the mesh is specified simply through the two global knot vectors $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$. As the LR mesh cannot be defined by global knot vectors, it has to be defined by its N_m knotline segments and their multiplicities. Each knotline segment is defined by a start point and an end point. When the multiplicities of all knotline segments in a given mesh are all set to 1, we will refer to it as the *grid*.

An LR mesh is a special kind of a mesh. The life of an LR mesh has two different stages:

1. The initial tensor-product construction
2. The subsequent local refinements

From the beginning, the LR mesh is constructed simply as a standard tensor-product mesh. An example is shown in Fig. 15.4a. The global knot vectors are $\boldsymbol{\eta}_1 = \boldsymbol{\eta}_2 = \{0, 0, 0, 1/5, 2/5, 3/5, 4/5, 1, 1, 1\}$. Using this tensor-product mesh as a starting point, the mesh is then refined by subsequently inserting knotline segments into it, such that the mesh remains a *box-partition*, i.e., consists of a collection of quadrilaterals throughout each refinement. Let us assume that we want to refine the highlighted box in Fig. 15.4a. First, Fig. 15.4b shows the result of a usual tensor-product refinement, i.e., when inserting one vertical and one horizontal knotline

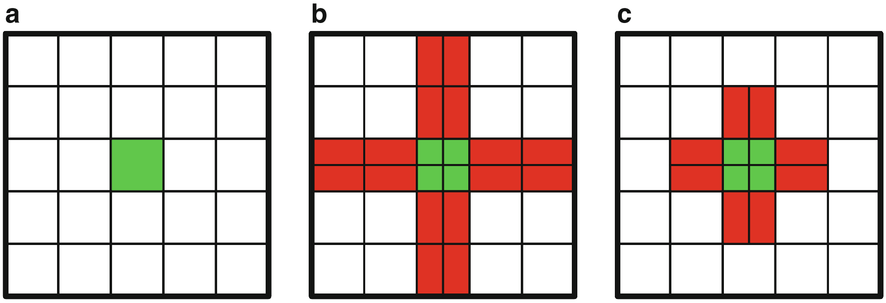


Fig. 15.4 Three different meshes: the initial tensor-product mesh (**a**), a globally refined tensor-product mesh (**b**), and a locally refined mesh (**c**). Multiplicities are three on the boundary and one elsewhere

segments through the box of interest and letting these extend all the way to the boundaries. This clearly identifies the problem with the tensor-product approach; along with the actual box of interest, *all* boxes towards the boundaries are also refined. The LR mesh, however, allows for much more local refinements. An example is shown in Fig. 15.4c. Here, we have inserted two short knotline segments, one vertical and one horizontal. As we shall see below, the knotline segments must be specified in such a way that each of them splits an LR B-spline. This is part of the reason why the knotline segments extend outside the highlighted box, and why the neighboring boxes are still refined. For consistency with the notation in different dimensions and settings, we usually refer to the knotline segments as *mesh-rectangles* and to the boxes as *elements*. These are central ingredients of an LR mesh.

15.3.3 LR B-Splines

With the LR mesh introduced, we now turn to the LR B-splines. An LR mesh gives rise to a number of LR B-splines, just like a tensor-product mesh gives rise to a number of tensor-product B-splines, cf. Fig. 15.3. By inserting local mesh-rectangles into an LR mesh, we enrich the space of B-splines living on it, and this in a much more local sense than by inserting global knots into a tensor-product mesh.

To illustrate this, let us return to the example from above, and assume that the polynomial degrees are $d_1 = d_2 = 2$. Figure 15.5a sketches the B-splines on the initial tensor-product mesh in Fig. 15.4a. To be more precise, the plot shows the Greville abscissae of the B-splines, which for any B-spline is just the average of the d_i central knots $\{\eta_{i,2}, \ldots, \eta_{i,d_i+1}\}$ in each parameter direction $i = 1, 2$, and thus a condensed way of visualizing the functions. By making the tensor-product refinements as in Fig. 15.4b, we end up with the B-splines sketched in Fig. 15.5b. The global nature of the refinement is again evident, as B-splines appear also

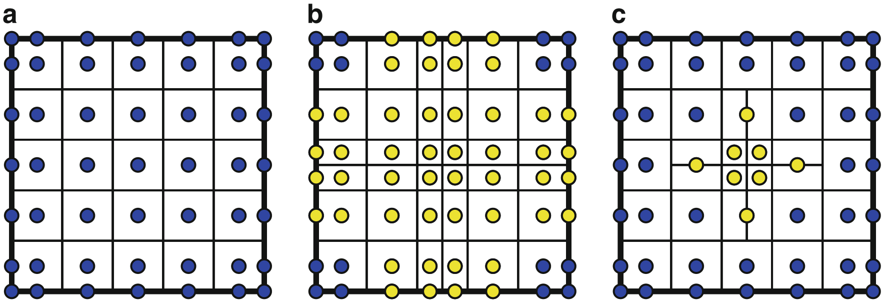


Fig. 15.5 Greville abscissae of all bi-quadratic B-splines on three different meshes: the initial tensor-product mesh (**a**), a globally refined tensor-product mesh (**b**), and a locally refined mesh (**c**). Multiplicities are three on the boundary and one elsewhere

away from the element of interest. On the other hand, by inserting the local mesh rectangles as in Fig. 15.4c, we obtain the LR B-splines shown in Fig. 15.5c. All new LR B-splines appear in close proximity of the element of interest. We mention in passing that, as the tensor-product meshes in Fig. 15.4a, b are indeed also LR meshes, the tensor-product B-splines in Fig. 15.5a, b are also LR B-splines.

But how do the new LR B-splines in Fig. 15.5c actually come about from the LR mesh in Fig. 15.4c? To answer this question, we consider Fig. 15.6. We first consider the insertion of the vertical mesh-rectangle. Remembering that all the tensor-product B-splines have support over 3×3 knot spans, we easily see that there are exactly three B-splines for which the mesh-rectangle traverses their entire support in the vertical direction, as indicated in Fig. 15.6a (left). These are the coarse functions that are to be refined. The resulting functions after this first refinement are shown in Fig. 15.6a (right). Note that each of the new LR B-splines has exactly the same underlying knot structure as a standard tensor-product B-spline. Next, we consider the insertion of the horizontal mesh-rectangle. Now, there are four LR B-splines for which the mesh-rectangle traverses its entire support in the vertical direction, as indicated in Fig. 15.6b (left). These are now the coarse functions that are to be refined. The resulting functions after this second refinement are shown in Fig. 15.6b (right). The order of insertion turns out to play no role. Thus, we may as well insert the horizontal mesh-rectangle first, and then vertical afterward; the final outcome will be the same.

LR B-splines possess many of the properties that standard tensor-product B-splines do. They are piecewise polynomial functions, they have compact support, and they form a partition of unity, i.e., they sum to one in all points, a property ensured through a simple scaling of each of the functions. Linear independence of a set of LR B-splines is not guaranteed per se. This is crucial when using them to solve equations like the Navier-Stokes equation. To ensure linear independence, the functions either have to be established through refinement schemes known a priori to result in linear independent LR B-splines, or they must be tested a posteriori through, e.g., a so-called peeling algorithm [8]. In Sect. 15.5 below, we shall return

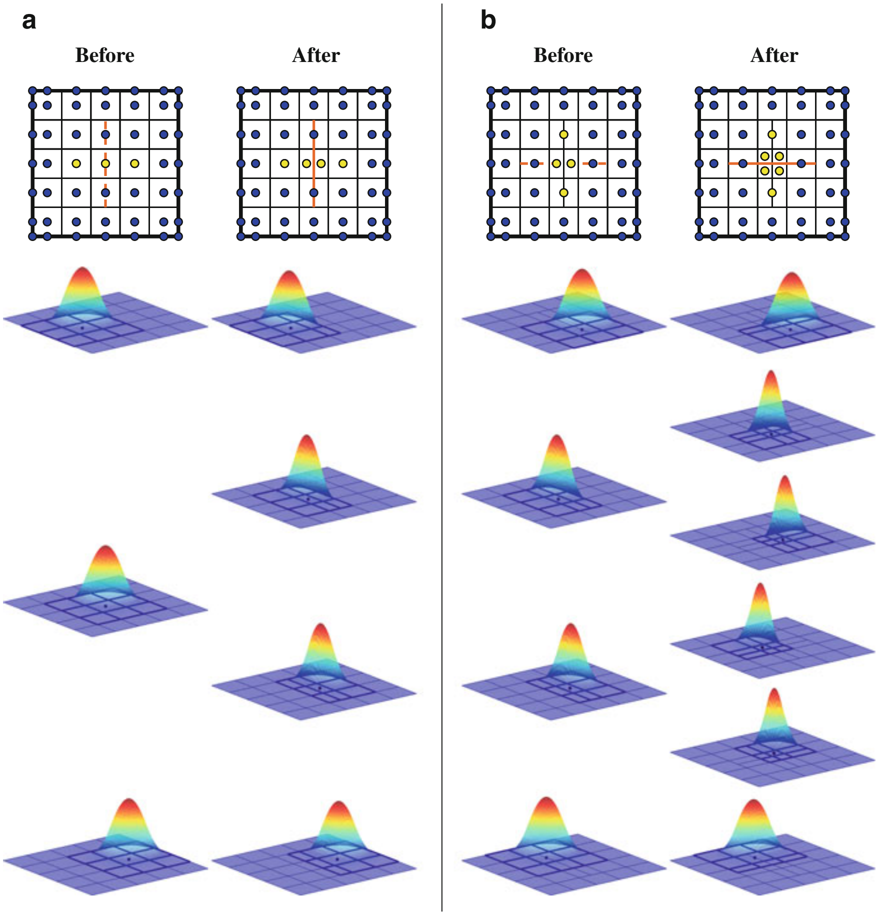


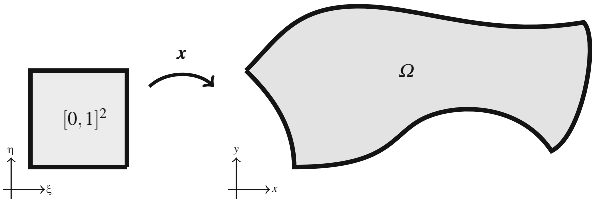
Fig. 15.6 B-splines before (*left*) and after (*right*) insertion of the vertical mesh rectangle (**a**) and the horizontal mesh rectangle (**b**). LR B-splines with Greville abscissae shown in *blue* are unaffected by the insertion, whereas LR B-splines with Greville abscissae shown in *yellow* are removed or inserted as a result of the insertion

to the construction of LR B-splines in the context of approximation of flow pressure and velocities for solving the Navier-Stokes equations.

15.4 Isogeometric Analysis

In this section, we outline the fundamentals of how to solve the Navier-Stokes and continuity equations (15.1) in an isogeometric framework based on LR B-splines. This essentially involves three ingredients: a parametrization of the geometry, a

Fig. 15.7 A parametrization of the flow domain Ω



discretization of the flow variables for the Galerkin projection, and the weak form of governing equations.

First, we construct a parametrization of the flow domain Ω, as sketched in Fig. 15.7. We take the parameter domain $\hat{\Omega}$ as the unit square and use bivariate LR B-splines as basis functions. The parametrization $\boldsymbol{x} : [0,1]^2 \rightarrow \mathbb{R}^2$ reads:

$$\boldsymbol{x}(\xi_1, \xi_2) = \sum_{i=1}^{N_g} \underline{\boldsymbol{x}}_i \mathcal{P}_i^g(\xi_1, \xi_2), \tag{15.5}$$

where $\underline{\boldsymbol{x}}_i$ are the control points, $\mathcal{P}_i^g$ are the LR B-splines, N_g is the number of LR B-splines and control points, and the superscript g indicates that the functions refer to the geometry parametrization.

Next, we seek approximations of the velocity $\boldsymbol{u}_h : [0,1]^2 \rightarrow \mathbb{R}^2$ and pressure $p_h : [0,1]^2 \rightarrow \mathbb{R}$ as linear combinations of LR B-splines, just like the geometry representation in Eq. (15.5) above:

$$\boldsymbol{u}_h(\xi_1, \xi_2) = \sum_{i=1}^{N_u} \underline{\boldsymbol{u}}_i \mathcal{P}_i^u(\xi_1, \xi_2)\,, \qquad p_h(\xi_1, \xi_2) = \sum_{i=1}^{N_p} \underline{p}_i \mathcal{P}_i^p(\xi_1, \xi_2). \tag{15.6}$$

Here, $\mathcal{P}_i^u$ and $\mathcal{P}_i^p$ denote the LR B-spline basis functions for the velocity and pressure, N_u and N_p are the number of velocity and pressure B-splines, while $\underline{\boldsymbol{u}}$ and $\underline{p}$ are the unknown control variables for the velocity and pressure, respectively. For simplicity, we discretize the two components of the velocity identically. The approximations in Eqs. (15.6) are defined in parameter space, whereas the governing equations (15.1) are posed in physical space. To evaluate the pressure in physical space $p : \Omega \rightarrow \mathbb{R}$, we use the inverse of the geometry parametrization as $p \circ \boldsymbol{x}^{-1}$. To evaluate the velocity in physical space $\boldsymbol{u} : \Omega \rightarrow \mathbb{R}^2$, we simply map each component as a scalar $\boldsymbol{I}\boldsymbol{u} \circ \boldsymbol{x}^{-1}$, where $\boldsymbol{I}$ is the identity map. Note here that, with abuse of notation, we use p and $\boldsymbol{u}$ to denote the pressure and the velocity, respectively, both over the physical space and over the parameter space. In Sect. 15.5 below, we describe in greater detail how to construct the LR B-spline discretizations of the pressure and velocity fields.

Finally, we cast the governing equations (15.1) and (15.2) into their weak form, which reads: find $(\boldsymbol{u}, p)$ with $\boldsymbol{u} = \boldsymbol{g}$ on $\partial\Omega$ and $\int_\Omega p \, \mathrm{d}\boldsymbol{x} = 0$ such that

$$0 = \int_\Omega \Big(\big(\frac{1}{\mathrm{Re}} \nabla v_k + v_k \, \boldsymbol{u} \big) \cdot \nabla u_k - \big(p \nabla v_k + v_k \, \boldsymbol{f} \big) \cdot \boldsymbol{e}_k \Big) \, \mathrm{d}\boldsymbol{x} , \quad k = 1, 2 , \tag{15.7a}$$

$$0 = \int_\Omega q \, (\nabla \cdot \boldsymbol{u}) \, \mathrm{d}\boldsymbol{x} \tag{15.7b}$$

for all $(\boldsymbol{v}, q)$ with $\boldsymbol{v} = \boldsymbol{0}$ on $\partial\Omega$, where we have used integration by parts in the derivation. Here, $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ are the standard Cartesian basis vectors, and the functions p and q must be square-integrable, while $\boldsymbol{u}$ and $\boldsymbol{v}$ as well as all their first-order derivatives must be square-integrable.

By using the LR B-splines approximations (15.6) as test and weight functions in the weak equations (15.7), and pulling the integrals back to the parameter domain based on the parametrization (15.5), a non-linear system of equations of the form $\boldsymbol{M}(\boldsymbol{U})\,\boldsymbol{U} = \boldsymbol{F}$ may be obtained:

$$\begin{bmatrix} \frac{1}{\mathrm{Re}} \boldsymbol{K} + \boldsymbol{C}(\underline{\boldsymbol{u}}) & \boldsymbol{0} & -\boldsymbol{G}_1^T \\ \boldsymbol{0} & \frac{1}{\mathrm{Re}} \boldsymbol{K} + \boldsymbol{C}(\underline{\boldsymbol{u}}) & -\boldsymbol{G}_2^T \\ \boldsymbol{G}_1 & \boldsymbol{G}_2 & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{u}}_1 \\ \underline{\boldsymbol{u}}_2 \\ \underline{\boldsymbol{p}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \\ \boldsymbol{0} \end{bmatrix} , \tag{15.8}$$

where

$$\boldsymbol{K}_{i,j} = \int\limits_{[0,1]^2} \nabla^T \mathcal{P}_i^u \, \boldsymbol{J}^{-1} \, \boldsymbol{J}^{-T} \, \nabla \mathcal{P}_j^u \, \det\big(\boldsymbol{J}\big) \, \mathrm{d}\boldsymbol{\xi} , \tag{15.9a}$$

$$\boldsymbol{C}_{i,j}(\boldsymbol{u}) = \int\limits_{[0,1]^2} \mathcal{P}_i^u \, \boldsymbol{u}^T \, \boldsymbol{J}^{-T} \, \nabla \mathcal{P}_j^u \, \det\big(\boldsymbol{J}\big) \, \mathrm{d}\boldsymbol{\xi} , \tag{15.9b}$$

$$\boldsymbol{G}_{k_{i,j}} = \int\limits_{[0,1]^2} \mathcal{P}_i^p \, \boldsymbol{e}_k^T \, \boldsymbol{J}^{-T} \, \nabla \mathcal{P}_j^u \, \det\big(\boldsymbol{J}\big) \, \mathrm{d}\boldsymbol{\xi} , \quad k = 1, 2 \tag{15.9c}$$

$$\boldsymbol{F}_{k_i} = \int\limits_{[0,1]^2} \mathcal{P}_i^u \, \boldsymbol{e}_k^T \, \boldsymbol{f} \, \det\big(\boldsymbol{J}\big) \, \mathrm{d}\boldsymbol{\xi} , \quad k = 1, 2 \tag{15.9d}$$

where $\boldsymbol{J}_{i,j} := \partial x_i / \partial \xi_j$ is the Jacobian matrix of the parametrization (15.5).

To solve the governing partial differential equations (15.1) using LR B-spline based isogeometric analysis, we thus need to solve the system of algebraic equations (15.8). To do this, we evaluate the integrals in (15.9) using Gaussian quadrature, and use an iterative Newton-Raphson solver. The Dirichlet boundary conditions (15.2a) on the velocity are enforced strongly, perhaps only in an approximative sense, while the additional condition (15.2b) on the mean pressure is imposed weakly through a least-square approach.

15.5 Flow Discretizations

We now proceed to introduce two families of discretizations of the pressure and velocity fields based on LR B-splines, thus substantiating the flow approximations (15.6) introduced above. We refer to these as the *Taylor-Hood* and *multigrid* families.

As indicated in Sect. 15.3, LR B-splines are characterized by a high degree of flexibility. In loose terms, we can play around with the two polynomial degrees, the two vectors of unique, global knots, and the multiplicities of each of the small mesh-rectangles along each of the global knots. When discretizing the flow variables, however, we narrow the scope slightly. As explained below, we choose to characterize the LR B-spline flow discretizations simply through one polynomial degree d, from which we then specify the polynomial degrees, regularities (smoothnesses) α, and levels of refinement r of both the pressure and the velocity fields.

The construction of both the Taylor-Hood and the multigrid flow discretizations comprises the same two stages as outlined in Sect. 15.3:

1. The tensor-product initialization
2. The subsequent local refinements

For now, we assume that a tensor-product spline representation $\boldsymbol{x}$ of the geometry is provided to us as input. We shall relax this assumption later.

In the initialization of both the Taylor-Hood and the multigrid discretization, we construct tensor-product spline representations of the velocity $\boldsymbol{u}_h$ and the pressure p_h in the usual fashion [5,17]. For both $\boldsymbol{u}_h$ and p_h, we take the global knot vectors $\boldsymbol{\eta}_i$ to be open, i.e., the multiplicity of the first and last knots are $d_i + 1$, and we take all interior knots to have the same multiplicity $m_i < d_i$ for the parametric dimensions $i = 1, 2$. Furthermore, in order to limit the number of parameters, and thus simplify the notation in the following, we assume for both $\boldsymbol{u}_h$ and p_h that the degree and the regularity are the same in both parametric dimensions, i.e., $d_1 = d_2 = d$ and $\alpha_1 = \alpha_2 = \alpha$, although these assumptions are not strictly required.

From the given tensor-product spline representation $\boldsymbol{x}$ of the geometry, we now choose a degree d and construct the tensor-product spline discretization p_h of the pressure using the same grid as for the geometry $\boldsymbol{x}$, with the degree $d^p = d$, full regularity $\alpha^p = d - 1$, and no refinements $r^p = 0$. The assumption of full regularity is not strictly required, but again it limits the number of parameters.

Next, we construct the discretization $\boldsymbol{u}_h$ of the velocity from p_h through one of two approaches: In the Taylor-Hood approach, we increase the polynomial degree $d^u = d^p + 1$, fix the regularity $\alpha^u = \alpha^p = d - 1$ by increasing the knot multiplicity, and keep the refinement level $r^u = 0$. In the multigrid approach, we increase both the polynomial degree $d^u = d^p + 1$, the regularity $\alpha^u = \alpha^p + 1 = d$, and the refinement level through insertion of, say, one additional knot in each regular knot span, such that $r^u = 1$. These tensor-product initializations are illustrated in the top of Fig. 15.8 for both the Taylor-Hood and multigrid discretization.

With the tensor-product initialization in place, we turn to the subsequent local refinements. For both the Taylor-Hood and the multigrid discretizations, we base

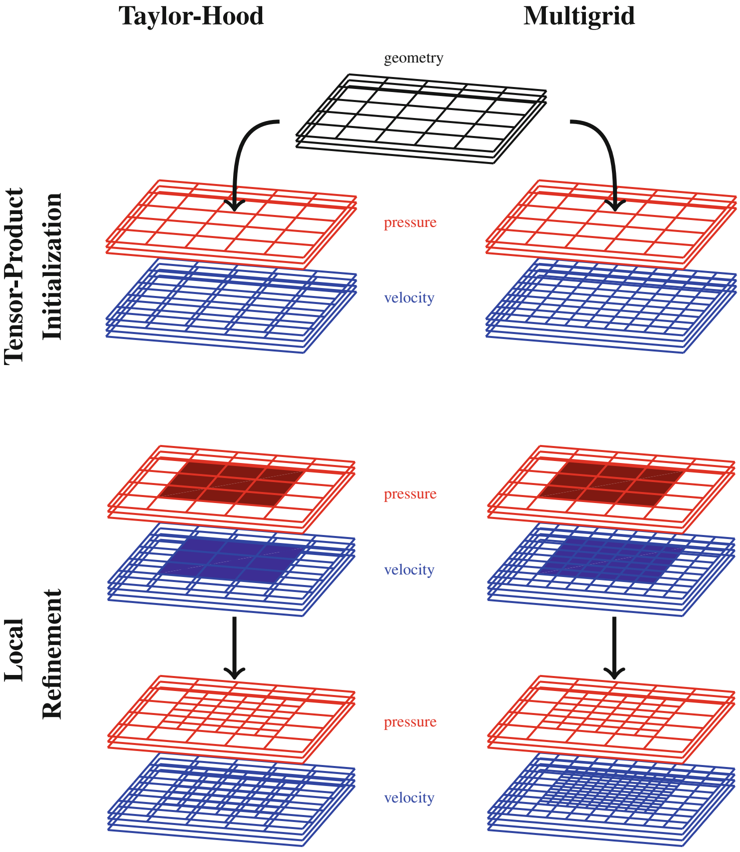


Fig. 15.8 Illustration of the tensor-product initialization (*top*) and subsequent local refinements (*bottom*) following the Taylor-Hood (*left*) and multigrid (*right*) approaches for degree $d = 2$

these on the structured mesh approach [16] with constant multiplicities $m^f = d^f - \alpha^f$ for both fields $f \in \{\boldsymbol{u}, p\}$. The idea, when refining according to the structured mesh approach, is to reason in terms of basis functions. First we choose a set of LR B-splines to refine, and then we insert new mesh-rectangles with the same multiplicity as the existing ones, in such a way that each regular span of mesh-rectangles whose two mesh-rectangles are both contained in the same LR B-spline within the specified set is split uniformly into n new spans of mesh-rectangles. Here, we will for simplicity use $n = 2$.

Now, when refining the flow discretization according to the structured mesh approach, we have both the LR B-splines of the *velocity* discretization and the LR B-splines of the *pressure* discretization to take into consideration. We choose to specify refinements for *both* the pressure and the velocity through a set of *pressure* LR B-splines. For both the pressure and the velocity discretization, we follow the structured mesh approach and base the insertions of mesh-rectangles on the support

Table 15.1 Characteristics of the two families of LR B-spline discretizations of the flow fields for given degree d

Discretization	Pressure			Velocity		
	Degree	Regularity	Refinement	Degree	Regularity	Refinement
Taylor-Hood	d	$d-1$	0	$d+1$	$d-1$	0
Multigrid	d	$d-1$	0	$d+1$	d	1

of a specified collection of pressure LR B-splines. This gives us *two* collections of mesh-rectangles; one collection is inserted in the *pressure* mesh, and one in the *velocity* mesh. These two collections of mesh-rectangles either differ in the number of mesh-rectangles or in the multiplicity of the mesh-rectangles, depending on the discretization in question. The number of pressure and velocity mesh-rectangles is dictated by the refinement level (e.g., N for both fields for the Taylor-Hood, and N and $n(N+1)-1$ for the pressure and velocity field, respectively, for the multigrid). The multiplicity of the pressure and velocity mesh-rectangles is given by the regularity (1 and 2, respectively, for the pressure and velocity fields for the Taylor-Hood, and 1 for both fields for the multigrid). These local refinements procedures are illustrated in the bottom of Fig. 15.8 for both the Taylor-Hood and the multigrid discretization, and their characteristics are summarized in Table 15.1.

It should be emphasized that the number of velocity elements for the multigrid discretization is larger than the number of velocity elements for the Taylor-Hood discretization by a factor of $(r+1)^2$. Since integrals in the matrices in Eq. (15.9) are evaluated based on elements, this makes the multigrid discretization computationally more expensive than the Taylor-Hood discretization.

We conclude by noting that the assumption of the geometry being discretized by a tensor-product spline can easily be relaxed. One obvious way to achieve this, while still ensuring that the geometry grid is contained within the velocity and pressure grids, is by assuming instead that the geometry is represented by an LR spline, that was initialized as a tensor-product spline of some degree based on open knot vectors with single interior knots, and whose subsequent refinements were all obtained using the structured mesh approach. In this case, we may construct the Taylor-Hood and multigrid flow discretization as before, except we must first set the initial tensor-product pressure discretization equal to the initial tensor-product geometry discretization, construct the tensor-product velocity discretization as before, and then go through the exact same steps of local refinements of the flow discretizations as for the final geometry discretization.

15.6 Numerical Examples

In this section, we test the LR B-spline flow discretizations introduced above in different numerical examples. Through these, we investigate the stability of the discretizations, we study their ability to reproduce an analytical solution, and we examine their performance on a standard benchmark problem.

15.6.1 Wall-Driven Annular Cavity: Stability

In the first example, we investigate the stability of the discretizations. We consider the problem outlined in Fig. 15.9a, in which a fluid is contained in an annular cavity, approximated by cubic B-splines. We are interested in the flow problem in the limit of small Reynolds numbers. Hence, we neglect the nonlinear term in the Navier-Stokes equation (15.1a), which then reduces to the Stokes equation. The sliding movement of the lower circular part of the boundary induces a rotating flow in the interior, while singularities in the pressure field form in the two lower corners, as indicated in Fig. 15.9b. Unstable discretizations manifest themselves qualitatively through spurious oscillations in the pressure field. Quantitatively, they violate the so-called inf–sup condition:

$$\inf_{p_h} \sup_{\boldsymbol{u}_h} \frac{\int_{\Omega} p_h \nabla \cdot \boldsymbol{u}_h \, \mathrm{d}\boldsymbol{x}}{\|p_h\|_{L^2} \|\boldsymbol{u}_h\|_{H^1}} \geq \beta > 0, \tag{15.10}$$

where the constant β is independent of the mesh resolution h.

In the following, we perform a series of numerical tests of whether the discretizations fulfill the inf–sup condition (15.10), i.e., whether they are stable or not, based on the rather degenerate problem sketched in Fig. 15.9. For each discretization, we refine a coarse mesh repeatedly, and estimate the value of β in each step [1, 6]. We follow two different schemes for choosing which LR B-splines to refine: by the first scheme, we refine the LR B-splines with support in one of the two lower corners, where the pressure singularities occur. By the second scheme, we refine a number of randomly chosen LR B-splines. Examples of the pressure grids produced by these two schemes are shown in Fig. 15.9c, d, respectively.

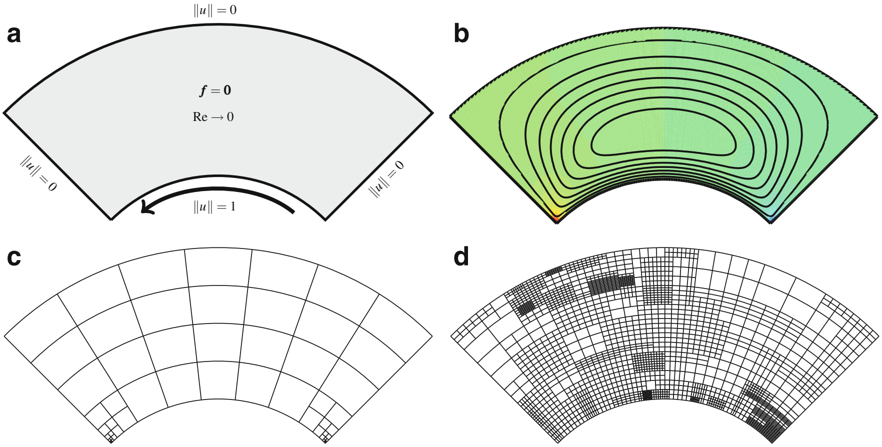


Fig. 15.9 Wall-driven annular cavity: problem setup (**a**), streamlines and pressure field (**b**), and pressure grid examples (**c** and **d**)

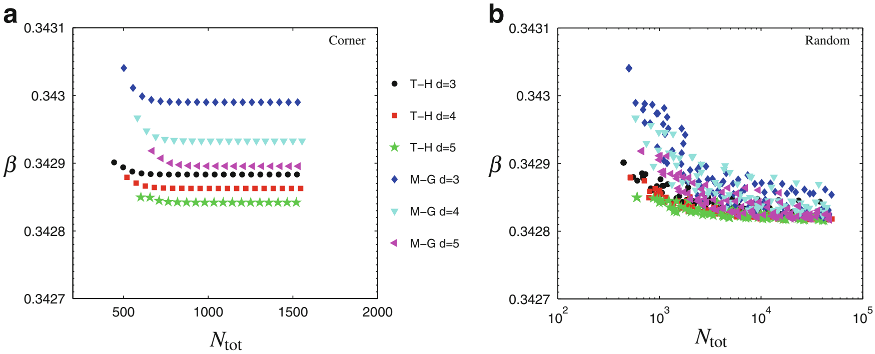


Fig. 15.10 Wall-driven annular cavity: Numerical estimates of the inf-sup "constant" β as a function of total number of analysis degrees of freedom N_{tot} for different discretizations (*T-H*: Taylor-Hood and *M-G* multigrid) based on corner function refinements (**a**) and random function refinements (**b**)

The estimated values of β for the Taylor-Hood and multigrid discretizations based on polynomial degrees 3–5, corresponding to degrees 3–5 for the pressure and 4–6 for the velocity, respectively, are shown in Fig. 15.10. A given discretization is said to pass the inf–sup test, if the estimated value of β does *not* tend to zero as the number of degrees of freedom is increased; if the value *does* tend to zero, the discretization fails the test. From these results, we are led to conclude that all the investigated discretizations among both the Taylor-Hood and the multigrid families pass the test. We emphasize that these conclusions are drawn on a (large but) finite number of numerical tests, and not on mathematical proofs. Furthermore, in addition to the Taylor-Hood and multigrid discretizations shown here, a discretization known to be unstable, based on a bi-quadratic pressure approximation and a bi-quartic velocity approximation, both fields having full regularity, was also tested and failed the test as expected.

15.6.2 Forced Wedge-Shaped Cavity: Manufactured Solution and Error Convergence

In this example, we study the ability of the discretizations to reproduce an analytical solution. Motivated by examples in [11, 20], we consider the problem outlined in Fig. 15.11a. A fluid is contained in the wedge-shaped region $\Omega = \{(x, y) \in \mathbb{R}^2 | \, 0 \leq x \leq 2, \, 0 \leq y \leq 1 + x^2/4\}$. We represent the domain exactly using quadratic B-splines. As target solution, we use the following velocity and pressure fields:

$$\begin{aligned} u^* = \frac{5}{532}\, \mathrm{e}^x\, x^2\, (x-2)^2\, y\, (4 - 4\, y + x^2) \\ \times (16 - 56\, y + 8\, x^2 + 40\, y^2 - 14\, yx^2 + x^4), \end{aligned} \qquad (15.11a)$$

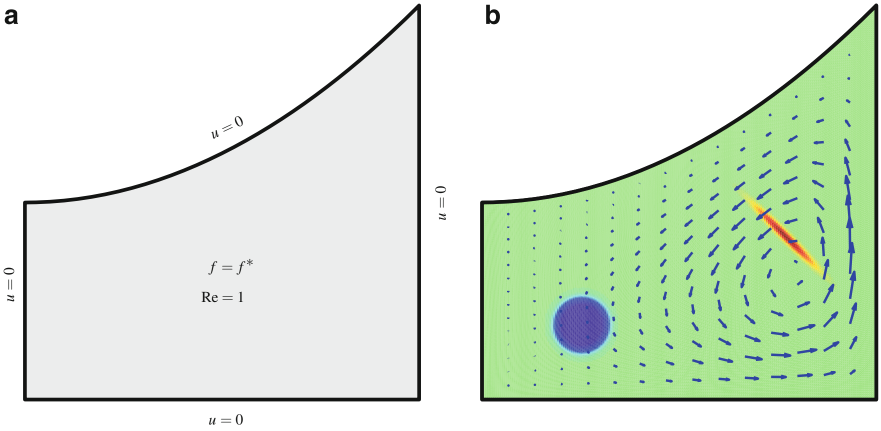


Fig. 15.11 Forced wedge-shaped cavity: problem setup (**a**) and velocity and pressure field (**b**)

$$\begin{aligned} v^* &= \frac{5}{266}\,\mathrm{e}^x\,x\,(x-2)\,y^2\,(4-4\,y+x^2)^2 \\ &\quad\times(16-8\,x-16\,y+12\,x^2+8\,xy-8\,x^3+4\,yx^2-x^4), \qquad (15.11\text{b}) \\ p^* &= \frac{1}{4}+\frac{1}{4}\,\tanh\Big(200\,(x-1/2)^2+200\,(y-3/8)^2-4\Big) \\ &\quad+\frac{1}{2}\,e^{-\left(5/2\,\sqrt{2}x-5/2\,\sqrt{2}y-5/2\right)^2-\left(25\,\sqrt{2}x+25\,\sqrt{2}y-250/3\right)^2}, \qquad (15.11\text{c}) \end{aligned}$$

as sketched in Fig. 15.11b. The velocity field is incompressible and fulfills the boundary conditions. By deriving the body force $\boldsymbol{f}^*$ through direct insertion into the Navier-Stokes equation, Eq. (15.11) is a manufactured analytical solution to the governing equations (15.1) and (15.2). As is evident from Fig. 15.11b, both the velocity and pressure fields exhibit phenomena that clearly call for local refinement.

To select which B-splines to refine, we follow two different approaches: a global approach and a local approach. By the global approach, we refine *all* functions, i.e., we are back in the tensor-product setting. By the local approach, we base the selection on the strong residual of the governing equations (15.1):

$$\boldsymbol{R}_h = \begin{pmatrix} (\boldsymbol{u}_h\cdot\nabla)u_h + \frac{\partial p_h}{\partial x} - \frac{1}{\mathrm{Re}}\Delta u_h + f_x \\ (\boldsymbol{u}_h\cdot\nabla)v_h + \frac{\partial p_h}{\partial y} - \frac{1}{\mathrm{Re}}\Delta v_h + f_y \\ \nabla\cdot\boldsymbol{u}_h \end{pmatrix}. \qquad (15.12)$$

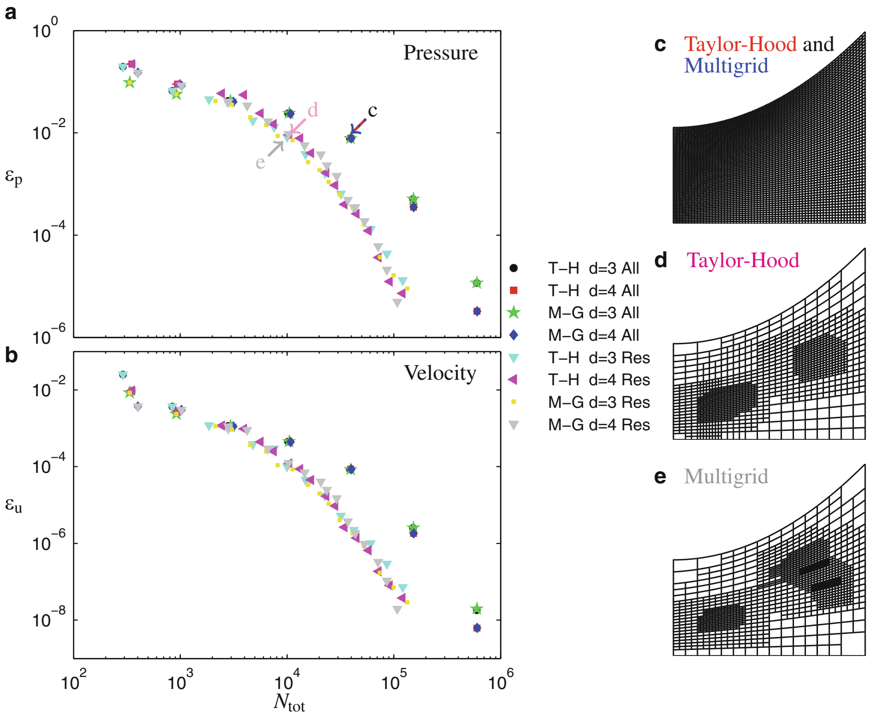


Fig. 15.12 Forced wedge-shaped cavity: Error convergence for the pressure (**a**) and velocity (**b**), and examples of the pressure grids produced by the global (**c**) and local refinement schemes for the Taylor-Hood (**d**) and the multigrid discretization (**e**) for degree $d = 4$

As error indicator, we use the L^2-norm of the residual vector $\boldsymbol{R}_h$. We integrate this on each element in the mesh, and for each (pressure) LR B-spline, we sum the errors from each of the elements in their support. Ordering the LR B-splines in a decreasing order according to their sum of errors, we refine the smallest number of LR B-splines that account for at least, say, 25 % of the total error. This may be seen as a *Dörfler* marking of LR B-splines.

In the following, we study how the global integrals of the L^2-norms of the errors on the pressure and the velocity fields behave as we refine the two discretization families based on each of the two refinement schemes. The results are shown in Fig. 15.12a, b for the pressure and the velocity fields, respectively, using polynomial degrees of 3 and 4. For any given number of degrees of freedom, the local refinement scheme is seen to yield significantly lower errors than the global refinement scheme, when comparing corresponding discretizations and degrees. For any given tolerance on the errors, hence, local refinement reduces the required number of degrees of freedom by up to an order of magnitude compared to global refinement. Comparing the Taylor-Hood and the multigrid discretizations for any given refinement strategy and polynomial degree, the two discretizations are seen to perform remarkably alike. As expected, we observe that higher polynomial degrees are associated with smaller errors. Also shown in Fig. 15.12c–e are examples of the pressure grids produced through global and local refinement of the Taylor-Hood and the multigrid discretizations, respectively, for degree $d = 4$. Although the globally refined grids in Fig. 15.12c have around four times as many degrees of freedom as the locally refined grids in Fig. 15.12d, e, all four produce results with similar errors. The local refinement scheme is seen to yield refinements in regions of strong gradients in the target pressure field, cf. Fig. 15.11b.

We mention at last that the quantitative aspects of these results of course depend on the specific problem, the error estimator, the refinement scheme, etc. We believe, however, that their qualitative aspects often will be the same.

15.6.3 Lid-Driven Square Cavity: Benchmark

In the last example, we examine the performance of the discretizations on a standard benchmark flow problem: the lid-driven square cavity. As outlined in Fig. 15.13a, the fluid in the square container is set in motion by the sliding movement of the lid of the container. The problem resembles the one in Sect. 15.6.1 above. However, although the geometry is simpler, the flow is now (weakly) turbulent with Re = 5,000. This introduces new challenges that we can test the LR B-spline flow discretizations against.

We solve the problem based on the LR Taylor-Hood and multigrid flow discretizations of degree $d = 3$ using the pressure grids shown in Fig. 15.13b, yielding a total of 22,515 and 22,785 degrees of freedom, respectively. The results of the computations are shown in Fig. 15.14. In Fig. 15.14a, the computed streamlines using the Taylor-Hood discretization clearly capture the counter-rotating eddies that are known to form in the NW, SE, and SW corners [10, 13]. The streamline pattern

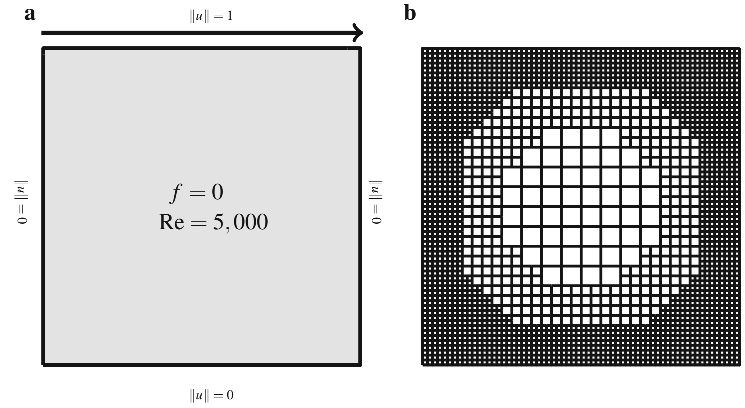


Fig. 15.13 The lid-driven square cavity: problem formulation (**a**) and pressure grids (**b**)

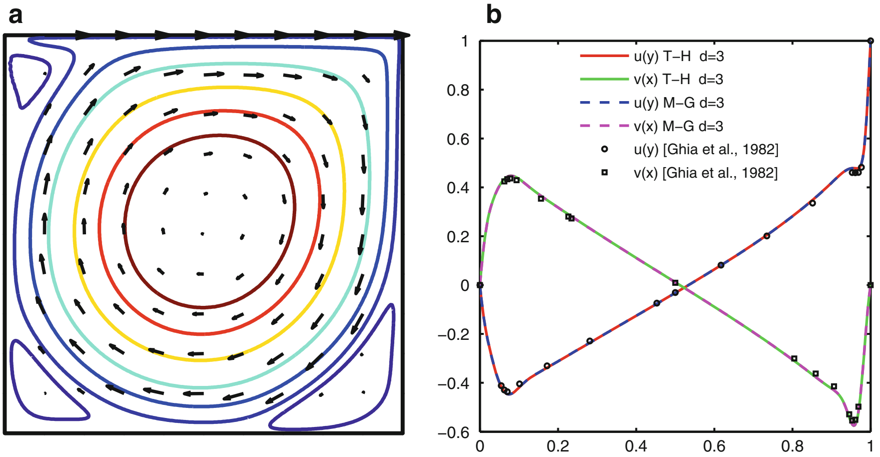


Fig. 15.14 The lid-driven square cavity: computed streamlines using the Taylor-Hood discretization (**a**) and normal velocity profiles through the two cavity center lines using both the Taylor-Hood (T-H) and multigrid (M-G) discretization (**b**)

looks the same for the multigrid discretization. In Fig. 15.14b, the normal velocity profiles through the cavity center lines computed using both the Taylor-Hood and the multigrid discretizations are seen to match very well with each other and with literature data [13].

Conclusions

The ability to achieve local refinement is crucial in all computer methods for flow problems. Locally Refinable B-splines represent a novel approach to local refinement within the context of isogeometric analysis. In this study, we have proposed two families of LR B-spline discretizations of the pressure and velocity fields for solving the mixed formulation of the steady-state, incompressible Navier-Stokes equations in two dimensions using isogeometric analysis. These LR flow discretizations represent direct extensions of well-known tensor-product flow discretizations, namely the Taylor-Hood and the multigrid discretizations. Through representative examples, we have performed a series of numerical investigations of the use of LR B-splines in isogeometric analysis of flow problems, including the stability of the discretizations, error convergence during refinement based on a manufactured solution, and benchmarking based on the lid-driven cavity problem.

Future investigations will hopefully reveal more insight into the properties of the flow discretizations. Their straightforward extensions to three dimensions should be studied. Extending the very promising but slightly more

(continued)

complicated Raviart-Thomas discretization to support local refinement is also of extreme interest, since this element satisfies the incompressibility condition exactly [11, 12]. Furthermore, efficient error estimators and refinement schemes should be studied to allow for efficient adaptive mesh refinement.

References

1. K. Bathe, The inf-sup condition and its evaluation for mixed finite element methods. Comput. Struct. **79**, 243–252 (2001)
2. Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, T.W. Sederberg, Isogeometric analysis using T-splines. Comput. Methods Appl. Mech. Eng. **199**, 229–263 (2010)
3. A. Bressan, Isogeometric regular discretization for the Stokes problem. IMA J. Numer. Anal. **31**(4), 1334–1356 (2011)
4. A. Bressan, Some properties of LR B-splines, Computer Aided Geometric Design, **30**(8), 778–794, (2013) doi: http://dx.doi.org/10.1016/j.cagd.2013.06.004
5. A. Buffa, C. de Falco, G. Sangalli, Isogeometric analysis: stable elements for the 2D Stokes equation. Int. J. Numer. Methods Fluids **65**(11–12), 1407–1422 (2011)
6. D. Chapelle, K. Bathe, The inf-sup test. Comput. Struct. **47**(4/5), 537–545 (1993)
7. J. Cottrell, T. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA* (Wiley, Chichester/Hoboken, 2009)
8. T. Dokken, K.F. Pettersen, T. Lyche, Locally refinable B-splines over box partions. Comput. Aided Geom. Des. **30**(3), 331–356 (2013)
9. M. Dörfel, B. Jüttler, B. Simeon, Adaptive isogeometric analysis by local h-refinement with T-splines. Comput. Methods Appl. Mech. Eng. **199**, 264–275 (2010)
10. E. Erturk, T. Corke, C. Gokcol, Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers. Int. J. Numer. Methods Fluids **48**, 747–774 (2005)
11. J.A. Evans, T.J.R. Hughes, Isogeometric divergence-conforming B-splines for the steady Navier-Stokes equations. Math. Models Methods Appl. Sci. **23**(8), 1421–1478 (2013)
12. J.A. Evans, T.J.R. Hughes, Isogeometric divergence-conforming B-splines for the unsteady Navier-Stokes equations. J. Comput. Fluids **241**, 141–167 (2013)
13. U. Ghia, K. Ghia, C. Shin, High-Re solution for incompressible flow using the Navier-Stokes equations and a multigrid method. J. Comput. Phys. **48**, 387–411 (1982)
14. C. Giannelli, B. Jüttler, H. Speleers, THB-splines: the truncated basis for hierarchical splines. Comput. Aided Geom. Des. **29**(7), 485–498 (2012)
15. T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Eng. **194**, 4135–4195 (2005)
16. K.A. Johannessen, T. Kvamsdal, T. Dokken, Isogeometric analysis using LR B-splines. Comput. Methods Appl. Mech. Eng. **269**, 471–514 (2014)
17. P.N. Nielsen, A.R. Gersborg, J. Gravesen, N.L. Pedersen, Discretizations in isogeometric analysis of Navier-Stokes flow. Comput. Methods Appl. Mech. Eng. **200**, 3242–3253 (2011)
18. M.A. Scott, X. Li, T.W. Sederberg, T.J.R. Hughes, Local refinement of analysis-suitable T-splines. Comput. Methods Appl. Mech. Eng. **213–216**, 206–222 (2012)
19. A.V. Vuong, C. Giannelli, B. Jüttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis. Comput. Methods Appl. Mech. Eng. **200**, 3554–3567 (2011)
20. G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Optimal analysis-aware parametrization of computational domain in isogeometric analysis, in *Advances in Geometric Modeling and Processing*, ed. by B. Mourrain, S. Schaeffer, G. Xu (Springer, Berlin, 2010), pp. 236–254

Index

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4