

# Isogeometric Approximation of Variational Problems for Shells

DISSERTATION

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Ricardo Perl

aus

Magdeburg

Bonn, Dezember 2015



---

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der  
Rheinischen Friedrich–Wilhelms–Universität Bonn  
am Institut für Numerische Simulation

1. Gutachter: Prof. Dr. Martin Rumpf  
2. Gutachter: Prof. Dr. Bert Jüttler  
Tag der Promotion: 12.04.2016  
Erscheinungsjahr: 2016



---

## Abstract

The interaction of *applied geometry* and *numerical simulation* is a growing field in the interplay of computer graphics, computational mechanics and applied mathematics known as *isogeometric analysis*. In this thesis we apply and analyze *Loop subdivision surfaces* as isogeometric tool because they provide great *flexibility* in handling surfaces of arbitrary topology combined with *higher order smoothness*. Compared with finite element methods, isogeometric methods are known to require far less degrees of freedom for the modeling of complex surfaces but at the same time the assembly of the isogeometric matrices is much more time-consuming. Therefore, we implement the isogeometric subdivision method and analyze the experimental convergence behavior for different quadrature schemes. The *mid-edge quadrature* combines robustness and efficiency, where efficiency is additionally increased via *lookup tables*. For the first time, the lookup tables allow the simulation with control meshes of arbitrary closed connectivity without an initial subdivision step, i.e. triangles can have more than one vertex with valence different from six.

*Geometric evolution problems* have many applications in material sciences, surface processing and modeling, bio-mechanics, elasticity and physical simulations. These evolution problems are often based on the *gradient flow* of a geometric energy depending on first and second fundamental forms of the surface. The isogeometric approach allows a conforming higher order spatial discretization of these geometric evolutions. To overcome a time-error dominated scheme, we combine *higher order space and time* discretizations, where the time discretization based on implicit Runge–Kutta methods. We prove that the *energy diminishes* in every time-step in the fully discrete setting under mild time-step restrictions which is the crucial characteristic of a gradient flow. The overall setup allows for a general type of fourth-order energies. Among others, we perform experiments for Willmore flow with respect to different metrics.

In the last chapter of this thesis we apply the *time-discrete geodesic calculus* in shape space to the *space of subdivision shells*. By approximating the squared Riemannian distance by a suitable energy, this approach defines a discrete path energy for a consistent computation of geodesics, logarithm and exponential maps and parallel transport. As approximation we pick up an *elastic shell energy*, which measures the deformation of a shell by membrane and bending contributions of its mid-surface. *Bézier curves* are a fundamental tool in computer-aided geometric design. We extend these to the subdivision shell space by generalizing the *de Casteljau* algorithm. The evaluation of Bézier curves depends on all input data. To solve this problem, we introduce *B-splines* and *cardinal splines* in shape space by gluing together piecewise Bézier curves in a smooth way. We show examples of quadratic and cubic Bézier curves, quadratic and cubic B-splines as well as cardinal splines in subdivision shell space.



# Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. Foundations</b>	<b>5</b>
2.1. Differential Geometry for Parametric Surfaces . . . . .	5
2.2. Subdivision Surfaces . . . . .	9
<b>3. Isogeometric Subdivision Method</b>	<b>23</b>
3.1. Spatial Discretization . . . . .	23
3.2. Weak Formulation . . . . .	25
3.3. Numerical Integration . . . . .	27
3.4. Numerical Experiments . . . . .	33
<b>4. Parametric Gradient Flows</b>	<b>39</b>
4.1. Excursion: Nested Variational Time Discretization for Willmore Flow . . . . .	42
4.2. Spatial Discretization . . . . .	51
4.3. Higher Order Implicit Runge–Kutta Methods in Time . . . . .	52
4.4. Robustness and Energy Decay . . . . .	56
4.5. Algorithm . . . . .	60
4.6. Numerical Results . . . . .	62
<b>5. Geodesics, Bézier Curves and Splines in Subdivision Shell Space</b>	<b>69</b>
5.1. Time-Discrete Geodesics Calculus in Shape Space . . . . .	70
5.2. Bézier Curves and Splines in Shape Space . . . . .	77
5.3. Subdivision Shell Space . . . . .	85
5.4. Numerical Implementation . . . . .	89
5.5. Numerical Results . . . . .	94
<b>A. Appendix</b>	<b>107</b>
A.1. Quartic Box Spline Shape Functions . . . . .	107
A.2. Additional Subdivision Matrices for Edge Evaluation and Quadrature Points . . . . .	107
A.3. First and Second Discrete Derivatives for Area and Willmore Functional and the Metrics	111
A.4. First and Second Discrete Derivatives for the Elastic Energy in Shape Space . . . . .	115
<b>Acknowledgments</b>	<b>127</b>
<b>Bibliography</b>	<b>129</b>





# 1. Introduction

In this thesis we develop numerical schemes for the isogeometric approximation of variational problems for shells. A *shell* is a three-dimensional solid whose thickness is very small when compared with other dimensions and can be represented by a mid-surface which is a two-dimensional embedded surface in  $\mathbb{R}^3$ . The geometric quantities of shells are characterized by its first *and* second fundamental forms. Consequently, the resulting Euler-Lagrange – or equilibrium – equations of the corresponding energy functionals are fourth-order partial differential equations. It is well-known from approximation theory, that these problems require suitable  $C^1$  resp.  $H^2$ -finite elements. With standard finite elements many problems arise in the construction of  $C^1$  -elements and hence, the numerical treatment of variational shell problems has a long history which is dominated by non-conforming approaches [21, 30].

During the last years, *isogeometric analysis* (IgA) [82, 37] was invented to unify the previously disjoint technologies of geometric design and numerical simulation. However, the isogeometric paradigm offers also many advantages in scientific computing that were hard to achieve with the finite element method. For instance, the discretization of higher-order geometric problems is straightforward with the isogeometric approach because of its smooth discretization spaces. Isogeometric surface representations are based on NURBS for quadrilateral meshes [58, 82] resp. box splines as their generalization to arbitrary polygonal meshes [40]. Unfortunately, spline surfaces are restricted to regular grids, so-called patches [58, 82]. There exist two dominating concepts for the construction of smooth surfaces of arbitrary topology. The most common approach in *computer-aided design* (CAD) for the last decades was to glue many different patches together to form so-called *multi-patch spline surfaces*. Modeling complex surfaces with multi-patch spline surfaces is time-consuming and complicated, because the gluing has to be done in the right way which requires user interaction. *Subdivision surfaces* creates higher-order smoothness by solving the  $N$ -sided hole filling problem [116]. More precisely, the subdivision approach generalizes the subdivision property near regular vertices to vertices of arbitrary connectivity so-called *extraordinary vertices*. When using the subdivision approach one can select ver-

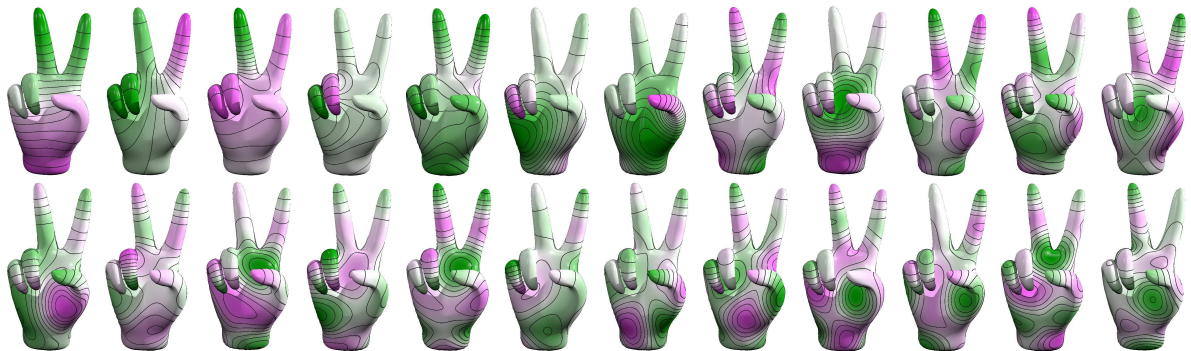


Figure 1.1.: Computation of the first 24 eigenfunctions of the Laplace-Beltrami operator computed on a complex surface using the isogeometric subdivision approach.

tices, edges or faces of the control mesh to manipulate the entire shape. Using standard additional split and extrude commands subdivision surfaces combine the power of parametric modeling with the flexibility of direct modeling. This way, the subdivision methodology changed the modeling paradigms in computer-aided design systems over the last years tremendously, see e.g. the *Freestyle* module of PTC Creo®, the *PowerSurfacing* add-on for SolidWorks® or *NX RealizeShape* of Siemens PLM® which employ the subdivision concept. Traditionally, subdivision schemes are widespread in geometry processing and computer graphics. Among the most popular subdivision schemes are the Catmull-Clark [26] and Doo-Sabin [46] schemes on quadrilateral meshes, and Loop's scheme on triangular meshes [97]. In this thesis we focus on Loop's subdivision scheme for closed meshes [97] but the developed numerical schemes can also be obtained using Catmull-Clark subdivision. Loop subdivision surfaces have been extensively studied in the literature, in particular regarding their smoothness [116], curvature integrability [117], (local) linear independence of their basis functions [112, 144], approximation power [5], and robust evaluation around extraordinary vertices [131]. Furthermore, Loop subdivision surfaces have been introduced as a finite element discretization technique for linear thin shell equations by Cirak and co-authors in [33]. Before the advent of isogeometric analysis, Cirak and co-authors [32] introduced subdivision surfaces for geometric modeling and mechanical simulation of thin shell structures.

**Isogeometric Subdivision Method** A bottleneck of isogeometric schemes is the efficient numerical integration of the discretized Euler-Lagrange systems. Indeed, the higher degree and multi-element support of the basis functions seriously affect the cost of robust numerical integration. Consequently, numerical quadrature in IgA is an active area of research [3, 83, 101, 100, 126]. Hence, one of the crucial steps in the implementation of the isogeometric subdivision method is the choice of numerical quadrature. The *computational cost*, *consistency*, *robustness* and the *observed order of convergence* are important parameters to evaluate the appropriateness of each numerical integration technique. Here, we focus on robust and efficient quadrature in the context of second and fourth-order PDEs. The robust integration is much more critical for subdivision surfaces than for NURBS because of the singular configuration at extraordinary vertices. In [33] the authors observe optimal convergence rates using the one-point barycenter quadrature rule in the energy-norm for all test cases. Our experiments suggest that the one-point barycenter quadrature rule can fail for fourth-order problems. Therefore, we propose the mid-edge quadrature, because it combines robustness and efficiency, where we additionally increase the efficiency via lookup tables. Furthermore, the natural parameterization introduced by Stam [131] only allows for triangles with at most one extraordinary vertex. Because the implementation is already very demanding the extension to triangles with more than one extraordinary vertex has not been done in the literature, yet. In these cases the subdivision control mesh has to be subdivided once to be parametrized by Stam's method. For the first time, our subdivision lookup tables allow the simulation with subdivision control meshes of arbitrary connectivity. This allows for a straightforward extension of existing subdivision modeling codes to simulations with subdivision surfaces for applications as performed in Figure 1.1.

**Parametric Gradient Flows** Because subdivision surfaces offer a great flexibility in modeling surfaces with  $C^1$ -smoothness they are ideally suited for the simulation of *geometric evolution problems* which have many theoretical and practical applications in mathematics and engineering sciences [18, 38, 52, 11, 7, 110, 20, 56, 138]. These evolutions are often based on the *gradient flow* of a geometric energy of the surface. We consider a general type of energies which involve the first and second derivatives of the surface leading to fourth-order partial differential equations for the corresponding

---

Euler-Lagrange equations. For the discretization with standard  $C^0$ -finite elements, these fourth-order equations have to be split in second-order problems which introduces additional degrees of freedom with possibly no geometric meaning. Furthermore, the splitting has to be performed for each fourth-order problem on its own. We review such a splitting for the case of anisotropic Willmore flow of curves and surfaces and present results for the evolution of curves. As a consequence of the splitting approach one can choose time steps of the size of the spatial grid size. In contrast, the direct, conforming discretization of fourth-order gradient flows with subdivision surfaces only allows time steps of the size of the squared spatial grid size. On the other hand, the discretization with subdivision elements results in a higher order spatial approximation. An effective implicit Euler time discretization would result in a time-error dominated scheme. Therefore, we apply a *higher order time discretization* based on *implicit Runge–Kutta methods* to balance the error terms. The crucial characteristic of a gradient flow is that the energy decreases in time. We prove that the *energy diminishing property* holds in the fully (time and space) discrete setting under mild time step restrictions. We apply the proposed scheme to a variety of geometric gradient flows, such as mean curvature and Willmore flow.

**Geodesics, Bézier Curves and Splines in Subdivision Shell Space** *Bézier curves* as well as *B-splines* and *cardinal splines* are fundamental tools in *computer-aided geometric design*. Applications range from vector graphics to CAD, from computer graphics or animation to computational engineering. All three concepts are linked by the *de Casteljau algorithm* [58] which is a robust and efficient approach to draw Bézier curves. In Euclidean space, the de Casteljau algorithm constructs polynomials by a weighted recursive combination of straight lines.

We consider the space of shells, where each shell is a point in space. Applying the direct computation of straight lines, Bézier curves and splines to shells as objects in Euclidean space results in many visual problems such as self-penetration of shells [78]. The reason is that the distance of two shells cannot effectively be measured by the Euclidean metric because it does not invoke the inherent structure of shells. Thus, the space of shells is often understood as a *Riemannian manifold* induced by a *Riemannian metric*. In the Riemannian setting, geodesics, i.e. paths of shortest distance, are the generalizations of straight lines in Euclidean space. Because the geodesic equations require a time stepping method which is usually not robust, Rumpf and Wirth [123] developed the *time-discrete geodesic calculus* for general shape spaces by introducing a suitable approximation of the squared Riemannian distance. Heeren and co-authors [78, 77, 79] applied the time-discrete geodesic calculus to the space of shells by defining an elastic deformation energy between two shells, which measures membrane and bending contributions. For the numerical implementation they applied a suitable discrete approximation of their previously defined continuous deformation energy using schemes from *discrete differential geometry*. Instead we will discretize their continuous deformation energy with subdivision elements such that we obtain time-discrete geodesics in the subdivision shell space. With geodesics as generalization of straight lines we define Bézier curves in shape space as a result of a generalized de Casteljau algorithm. Because the evaluation depends on all input shapes of the Bézier curve, we construct B-splines in shape space by gluing together piecewise Bézier curves analogously to the Euclidean setting. Since both approaches do not interpolate all input shapes, we additionally define cardinal splines in shape space.

**Publications and collaborations** Parts of this thesis have already been published and resulted from the collaboration with other researchers. The content of the isogeometric subdivision method in Section 2.2 and Chapter 3 was done in collaboration with Bert Jüttler, Angelos Mantzaflaris and Martin Rumpf and resulted in the preprint [86]. Section 4.1 is a summary of the joint work with Paola Pozzi

## *1. Introduction*

---

and Martin Rumpf on nested variational time discretization for anisotropic Willmore flow published in [110]. Finally, Chapter 4 arose from the collaboration with Dhia Mansour and Martin Rumpf, which has not yet been published but is in preparation [99].

## 2. Foundations

The aim of this thesis is to develop numerical schemes for higher order geometric PDEs. Therefore, we will recapitulate some basic definitions from differential geometry, where we follow [94, 45], as well as the definitions of Sobolev spaces [57]. Furthermore, we review Loop subdivision surfaces as a suitable  $H^2$ -conforming space discretization.

### 2.1. Differential Geometry for Parametric Surfaces

Let  $\mathcal{M}$  be a *two-dimensional surface* embedded in  $\mathbb{R}^3$ . Assume there is a mapping

$$\begin{aligned} x : \Omega &\rightarrow \mathcal{M} \subset \mathbb{R}^3 \\ \xi &\mapsto x(\xi) = p \end{aligned}$$

that assigns to all  $\xi$  from the *domain manifold*  $\Omega$  a point  $p \in \mathcal{M}$ . Then, we say  $x$  is a *parameterization* of  $\mathcal{M}$  and we write  $\mathcal{M} = \mathcal{M}[x]$ . Furthermore, we assume that  $\det(Dx(\xi)^T Dx(\xi)) \neq 0$  for all  $\xi \in \Omega$  where

$$Dx = (x_{,1} \quad x_{,2}) = \begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ x_{3,1} & x_{3,2} \end{pmatrix} = \begin{pmatrix} \frac{\partial x_1(\xi)}{\partial \xi_1} & \frac{\partial x_1(\xi)}{\partial \xi_2} \\ \frac{\partial x_2(\xi)}{\partial \xi_1} & \frac{\partial x_2(\xi)}{\partial \xi_2} \\ \frac{\partial x_3(\xi)}{\partial \xi_1} & \frac{\partial x_3(\xi)}{\partial \xi_2} \end{pmatrix} = \left( \frac{\partial x(\xi)}{\partial \xi_i} \right)_{i=1,2}$$

denotes the *Jacobian* of the parameterization  $x$ . In particular, this means that  $\mathcal{M}$  is a *regular surface* in  $\mathbb{R}^3$ .

Let  $S^2$  be the unit sphere in  $\mathbb{R}^3$ . The *Gauß map* is defined as the normal mapping

$$\begin{aligned} n : \mathcal{M} &\rightarrow S^2 \\ p &\mapsto n(p) = n, \end{aligned}$$

where  $n$  is the outer *normal* on the surface  $\mathcal{M}$  at point  $p \in \mathcal{M}$ . Using the Jacobian we can represent  $n$  in local coordinates of the parameterization  $x$

$$n = n(p) = n(x(\xi)) = \frac{x_{,1} \times x_{,2}}{|x_{,1} \times x_{,2}|},$$

where  $|p| = \sqrt{p^T p} = \sqrt{p \cdot p} = \sqrt{p_1^2 + p_2^2 + p_3^2}$  denotes the vector norm of  $\mathbb{R}^3$  and  $v \cdot w = v^T w$  the corresponding scalar product of vectors  $v$  and  $w$ . The tangent space  $\mathcal{T}_p \mathcal{M}$  is the two-dimensional linear subspace of  $\mathbb{R}^3$  that is orthogonal to the normal  $n = n(p)$  at a point  $p \in \mathcal{M}$ , i.e.

$$\mathcal{T}_p \mathcal{M} = \{v \in \mathbb{R}^3 \mid v \cdot n = 0\}.$$

## 2. Foundations

---

For a parameterization  $x$  we can identify the tangent space for a point  $x(\xi) = p \in \mathcal{M}$  by

$$\mathcal{T}_p\mathcal{M} = \mathcal{T}_{x(\xi)}\mathcal{M} = \left\{ \sum_{i=1}^2 v_i \frac{\partial x(\xi)}{\partial \xi_i} \in \mathbb{R}^3 \mid v = [v_1, v_2]^T \in \mathbb{R}^2 \right\} = \{Dx(\xi) \cdot v \mid v \in \mathbb{R}^2\}.$$

Now, we introduce the *first fundamental form*  $g_p$  of a surface  $\mathcal{M}$  at a point  $p \in \mathcal{M}$  by

$$g_p : \mathcal{T}_p\mathcal{M} \times \mathcal{T}_p\mathcal{M} \rightarrow \mathbb{R}, \quad g_p(U, W) := U \cdot W, \quad U, W \in \mathcal{T}_p\mathcal{M}$$

which allows us to measure on the surface. For two tangent vectors  $U, W \in \mathcal{T}_p\mathcal{M}$ , using the basis representation  $U = Dx \cdot u$  and  $W = Dx \cdot w$  for  $u, w \in \mathbb{R}^2$ , we can express the first fundamental form in local coordinates of  $\Omega$  (called *pull-back*) by

$$g_p(U, W) = u^T (Dx^T Dx) w = u^T g \cdot w = g_\xi(u, w).$$

In the following we will often write  $g$  instead of  $g_p$  or  $g_\xi$  and assume that the context is clear, but most of the time we will refer to the bilinear form  $g = (g_{ij})_{i,j=1,2} = (x_{,i} \cdot x_{,j})_{i,j=1,2}$ . Furthermore, we denote the inverse of  $g$  by  $g^{-1}$ .

Now, we want to introduce the notion of *curvature* of a regular manifold  $\mathcal{M}$ , i.e. we want measure how the normal field  $n$  changes on the surface  $\mathcal{M}$ . Therefore, we assume that  $n$  is a smooth normal field, i.e.  $Dn = Dn(p) = (n_{,1} \ n_{,2})$  exists for all  $p \in \mathcal{M}$  and that  $n$  is at least continuous which implies that the surface  $\mathcal{M}$  is orientable. Then, using the relation  $n \cdot n = 1$ , we obtain

$$n \cdot Dn = 0$$

which implies that  $n_{,1}$  and  $n_{,2}$  lie in the tangent space  $\mathcal{T}_p\mathcal{M}$ . This motivates the definition of a second scalar product on the tangent space  $\mathcal{T}_p\mathcal{M}$  known as the *second fundamental form*  $b_p : \mathcal{T}_p\mathcal{M} \times \mathcal{T}_p\mathcal{M} \rightarrow \mathbb{R}$  with

$$b_p(V, U) = -(Dn \cdot v) \cdot (Dx \cdot u) = v^T (-Dn^T Dx) \cdot u = v^T b u = b_\xi(v, u),$$

where  $V = Dn \cdot v \in \mathcal{T}_p\mathcal{M}$  and  $U = Dx \cdot u \in \mathcal{T}_p\mathcal{M}$ . To get an easier representation of the matrix  $b$  we compute the derivative of  $n^T Dx = 0$  and obtain for  $i, j = 1, 2$  that

$$b_{ij} = -n_{,i} \cdot x_{,j} = n \cdot x_{,ij},$$

where

$$D^2x = \left( \frac{\partial^2 x(\xi)}{\partial \xi_j \partial \xi_i} \right)_{i,j=1,2}$$

denotes the *Hessian* of  $x$ .

Because  $n \cdot Dn = 0$  and  $n \cdot Dx = 0$ , i.e.  $Dn$  and  $Dx$  lie in same subspace, there exists a linear map  $s_p : \mathcal{T}_p\mathcal{M} \rightarrow \mathcal{T}_p\mathcal{M}$  such that

$$b_q(V, U) = g_p(s_p \cdot V, U), \quad \forall V, U \in \mathcal{T}_p\mathcal{M}$$

known as *embedded Weingarten map* or *embedded shape operator*. If we further assume that  $\mathbf{s}_p \cdot \mathbf{n} = 0$ ,  $\mathbf{s}_p$  has the unique representation

$$\mathbf{s}_p = -Dn g^{-1} Dx^T = Dx g^{-1} b g^{-1} Dx^T,$$

where we use that

$$b_q(V, U) = -(Dn \cdot v) \cdot (Dx \cdot u) = (\mathbf{s}_p Dx \cdot v) \cdot (Dx \cdot u) = g_p(\mathbf{s}_p \cdot V, U), \quad \forall V, U \in \mathcal{T}_p \mathcal{M},$$

with

$$-Dn = \mathbf{s}_p Dx.$$

In addition to  $\mathbf{s}_q$  we can define  $\mathbf{s}_\xi$  as the *Weingarten map* on the corresponding tangent space in  $\mathbb{R}^2$ . Then,  $\mathbf{s}_\xi$  has to fulfill the following condition

$$b_\xi(v, u) = g_\xi(\mathbf{s}_\xi \cdot v, u), \quad \forall v, u \in \mathbb{R}^2,$$

such that

$$\mathbf{s}_\xi = b \cdot g^{-1}$$

respectively

$$\mathbf{s}_p = Dx g^{-1} \cdot \mathbf{s}_\xi Dx^T.$$

Let us remark that the main difference between  $\mathbf{s}_p$  and  $\mathbf{s}_\xi$  is that  $\mathbf{s}_p$  is independent of the parameterization and  $\mathbf{s}_\xi$  depends on the parameterization. In many cases one can directly work with  $\mathbf{s}_\xi$  (e.g. because it is faster), but since we work with parameterizations it is important to define all quantities with respect to  $\mathbf{s}_q$ . Then, we can compute the corresponding representation in  $\mathbf{s}_\xi$ , e.g. the Frobenius norm of  $\mathbf{s}_p$  is not equal to the Frobenius norm of  $\mathbf{s}_\xi$ . In particular,

$$\begin{aligned} |\mathbf{s}_q|_{\mathbb{F}}^2 &= \text{tr}(\mathbf{s}_p^T \mathbf{s}_p) = \text{tr} \left( (Dx g^{-1} \cdot \mathbf{s}_\xi Dx^T)^T (Dx g^{-1} \cdot \mathbf{s}_\xi Dx^T) \right) \\ &= \text{tr} \left( (g^{-1} \cdot \mathbf{s}_\xi Dx^T)^T (\mathbf{s}_\xi Dx^T) \right) \\ &= \text{tr}(\mathbf{s}_\xi \cdot \mathbf{s}_\xi) \neq \text{tr}(\mathbf{s}_\xi^T \cdot \mathbf{s}_\xi) = |\mathbf{s}_\xi|_{\mathbb{F}}^2, \end{aligned}$$

where  $|A|_{\mathbb{F}} = \sqrt{\sum_{i=1}^n \sum_{j=1}^m A_{ij}^2}$  denotes the Frobenius norm of a matrix  $A \in \mathbb{R}^{n \times m}$ . Furthermore,  $\mathbf{s}_p$  is symmetric and has the eigenvalues  $\kappa_1, \kappa_2$  and 0, where  $\kappa_1$  and  $\kappa_2$  denote the *principal curvatures* of the surface  $\mathcal{M}$  at the point  $p$ , where  $\mathbf{s}_\xi$  has the eigenvalues  $\kappa_1$  and  $\kappa_2$ . In both cases these eigenvalues exist because  $\mathbf{s}_p$  and  $\mathbf{s}_\xi$  are self-adjoint with respect to the metric  $g_p$  respectively  $g_\xi$ , i.e.  $g_p(\mathbf{s}_p \cdot V, U) = g_p(V, \mathbf{s}_p \cdot U)$  for  $U, V \in \mathcal{T}_p \mathcal{M}$ . Now, we have everything at hand to introduce the *mean curvature*  $\mathbf{h}$  at a point  $x(\xi) = p \in \mathcal{M}$  by

$$\mathbf{h} = \text{tr}(\mathbf{s}_p) = \text{tr}(\mathbf{s}_\xi) = \kappa_1 + \kappa_2$$

## 2. Foundations

---

(here, the mean curvature is the sum of the eigenvalues instead of the mean).

Let  $u : \mathcal{M} \rightarrow \mathbb{R}$  be a smooth scalar function defined on  $\mathcal{M}$ . Then the gradient of  $u$  on  $\mathcal{M}$

$$\nabla_{\mathcal{M}} u(x) = \begin{pmatrix} \nabla_{\mathcal{M}}^1 u(x) \\ \nabla_{\mathcal{M}}^2 u(x) \\ \nabla_{\mathcal{M}}^3 u(x) \end{pmatrix} = \begin{pmatrix} (Dx_1^T g^{-1} \nabla_{\xi}(u \circ x))(\xi) \\ (Dx_2^T g^{-1} \nabla_{\xi}(u \circ x))(\xi) \\ (Dx_3^T g^{-1} \nabla_{\xi}(u \circ x))(\xi) \end{pmatrix} = (Dx g^{-1} \nabla_{\xi}(u \circ x))(\xi)$$

is known as the *tangential gradient* or *surface gradient*. Here,  $\nabla_{\xi} \tilde{u}(\xi) = \begin{pmatrix} \frac{\partial \tilde{u}(\xi)}{\partial \xi_1} \\ \frac{\partial \tilde{u}(\xi)}{\partial \xi_2} \end{pmatrix}$  denotes the gradient of a scalar function  $\tilde{u} : \Omega \rightarrow \mathbb{R}$  on  $\Omega$ . Furthermore,  $\operatorname{div}_{\xi} \tilde{v} = \frac{\partial \tilde{v}_1}{\partial \xi_1} + \frac{\partial \tilde{v}_2}{\partial \xi_2}$  denotes the divergence of a vector field  $\tilde{v} : \Omega \rightarrow \mathbb{R}^2$  on  $\Omega$ . Then, we define the *Laplace-Beltrami* operator by

$$\Delta_{\mathcal{M}} u(x) = \nabla_{\mathcal{M}} \cdot \nabla_{\mathcal{M}} u(x) = \operatorname{div}_{\mathcal{M}} \nabla_{\mathcal{M}} u(x) = \frac{1}{\sqrt{\det g}} \operatorname{div}_{\xi} \left( \sqrt{\det g} g^{-1} \nabla_{\xi}(u \circ x) \right) (\xi).$$

Both definitions can be generalized to mappings  $y : \Omega \rightarrow \mathbb{R}^3$  by

$$\nabla_{\mathcal{M}} y = \begin{pmatrix} (\nabla_{\mathcal{M}} y_1)^T \\ (\nabla_{\mathcal{M}} y_2)^T \\ (\nabla_{\mathcal{M}} y_3)^T \end{pmatrix} \quad \text{resp.} \quad \Delta_{\mathcal{M}} y = \begin{pmatrix} \Delta_{\mathcal{M}} y_1 \\ \Delta_{\mathcal{M}} y_2 \\ \Delta_{\mathcal{M}} y_3 \end{pmatrix}.$$

Let us conclude this section by introducing the Sobolev spaces  $W^{m,p}(\mathcal{M}, \mathbb{R})$  for functions  $u : \mathcal{M} \rightarrow \mathbb{R}$  and  $W^{m,p}(\mathcal{M}, \mathbb{R}^3)$  of parameterizations  $x$ . First, let us define the corresponding  $L^p$  spaces for the function itself, its surface gradient and surface Hessian by

$$\|u\|_{L^p(\mathcal{M}, \mathbb{R})} = \left( \int_{\mathcal{M}} |u|^p \, da \right)^{\frac{1}{p}},$$

$$\|\nabla_{\mathcal{M}} u\|_{L^p(\mathcal{M}, \mathbb{R})} = \left( \int_{\mathcal{M}} |\nabla_{\mathcal{M}} u|^p \, da \right)^{\frac{1}{p}},$$

where  $|\nabla_{\mathcal{M}} u|^p = \sum_k (\nabla_{\mathcal{M}}^k u)^p$ , and

$$\|\nabla_{\mathcal{M}}^2 u\|_{L^p(\mathcal{M}, \mathbb{R})} = \left( \int_{\mathcal{M}} |\nabla_{\mathcal{M}}^2 u|^p \, da \right)^{\frac{1}{p}},$$

where  $|\nabla_{\mathcal{M}}^2 u|^p = \sum_{i,j} (\nabla_{\mathcal{M}}^i \nabla_{\mathcal{M}}^j u)^p$  and  $da = \sqrt{\det g} \, d\xi$  denotes the area element. Then,

$$\|u\|_{W^{0,p}(\mathcal{M}, \mathbb{R})} = \|u\|_{L^p(\mathcal{M}, \mathbb{R})},$$

$$\|u\|_{W^{1,p}(\mathcal{M}, \mathbb{R})} = \left( \|u\|_{L^p(\mathcal{M}, \mathbb{R})}^p + \|\nabla_{\mathcal{M}} u\|_{L^p(\mathcal{M}, \mathbb{R})}^p \right)^{\frac{1}{p}},$$

$$\|u\|_{W^{2,p}(\mathcal{M}, \mathbb{R})} = \left( \|u\|_{L^p(\mathcal{M}, \mathbb{R})}^p + \|\nabla_{\mathcal{M}} u\|_{L^p(\mathcal{M}, \mathbb{R})}^p + \|\nabla_{\mathcal{M}}^2 u\|_{L^p(\mathcal{M}, \mathbb{R})}^p \right)^{\frac{1}{p}},$$



and

$$\|u\|_{H^m(\mathcal{M},\mathbb{R})} = \|u\|_{W^{m,2}(\mathcal{M},\mathbb{R})} \quad \forall m = 0, 1, 2.$$

The definitions can be generalized to mappings  $y : \Omega \rightarrow \mathcal{M}$  by

$$\|y\|_{L^p(\mathcal{M},\mathbb{R}^3)} = \left( \int_{\mathcal{M}} |y_1|^p + |y_2|^p + |y_3|^p \, da \right)^{\frac{1}{p}},$$

$$\|\nabla_{\mathcal{M}} y\|_{L^p(\mathcal{M},\mathbb{R}^3)} = \left( \int_{\mathcal{M}} |\nabla_{\mathcal{M}} y_1|^p + |\nabla_{\mathcal{M}} y_2|^p + |\nabla_{\mathcal{M}} y_3|^p \, da \right)^{\frac{1}{p}}$$

and

$$\|\nabla_{\mathcal{M}}^2 y\|_{L^p(\mathcal{M},\mathbb{R}^3)} = \left( \int_{\mathcal{M}} |\nabla_{\mathcal{M}}^2 y_1|^p + |\nabla_{\mathcal{M}}^2 y_2|^p + |\nabla_{\mathcal{M}}^2 y_3|^p \, da \right)^{\frac{1}{p}}.$$

All other definitions are analog to the scalar function case. Let us note that we write  $W^{m,p}$  instead of  $W^{m,p}(\mathcal{M}, \mathbb{R})$  respectively  $W^{m,p}(\mathcal{M}, \mathbb{R}^3)$  and call a function or parameterization in  $W^{m,p}$  if  $\|u\|_{W^{m,p}} < \infty$  respectively  $\|y\|_{W^{m,p}} < \infty$ .

## 2.2. Subdivision Surfaces

In this section we describe the  $H^2$ -conforming discretization based on Loop's subdivision scheme [97]. As already mentioned before, subdivision surfaces generalize the subdivision property of box spline surfaces (see [40] for a definition) to meshes of arbitrary topology. For a comprehensive introduction to subdivision methods in general we refer to [111, 128] and [25]. For the integration of subdivision methods in CAD systems, we refer to [4, 19] and the references therein.

Since their invention in the late 1970s (the first schemes were based on quadrilaterals [26, 46]) subdivision surfaces attracted much attention from researchers and designers because of their elegant and efficient way to produce visually appealing surfaces of arbitrary topology. But it took more than two decades before Reif [116] developed a unified framework to prove the conjectured  $C^1$ -property using the *characteristic map embedding* and Reif and Schröder [117] could show the  $H^2$ -property which is essential for higher order finite element simulations, which were first performed by Cirak, Ortiz and Schröder in [33] for linear thin shell equations. In [70, 31] the setup was extended to the nonlinear regime. For the linear thin shell problem described in [33], Green and co-authors [68] developed a multigrid preconditioned conjugate gradient algorithm based on the subdivision matrix which scales almost linearly to account for large scale engineering simulations.

In the following we introduce the refinement rules used in Loop's subdivision scheme [97] for surfaces, the corresponding subdivision matrix and their eigenvalue structure needed for the smoothness analysis. Furthermore, we describe how to construct the domain manifold and basis functions. Based on these we briefly discuss the concept of the *characteristic map* based on eigenfunctions and list the main results of the smoothness analysis. Finally, we discuss the evaluation of positions, tangents and curvatures using the *natural parameterization* introduced by Stam [131].

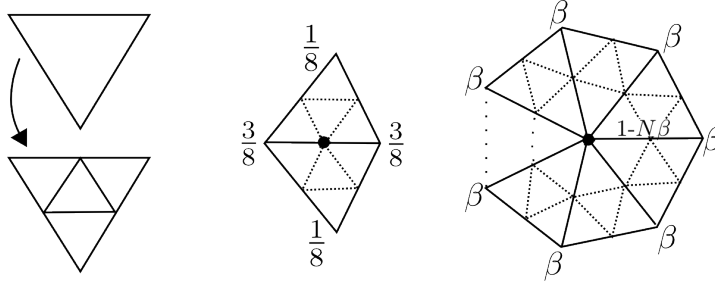


Figure 2.1.: Loop's original subdivision stencils. On the left we depict the topological splitting of a triangle into four smaller triangles called quadrisection. The corresponding edge and vertex rules are depicted in the middle respectively on the right where  $\beta = \beta(N) = \frac{1}{N} \left( \frac{5}{8} - \left( \frac{3}{8} + \frac{1}{4} \cos \left( \frac{2\pi}{N} \right) \right)^2 \right)$ .

**Subdivision matrix** Let us first clarify some notation. We denote by  $\mathcal{T}$  a *closed simplicial mesh* where  $\mathcal{I}_v \subset \mathbb{Z}$  denotes the *vertex index set*,  $\mathcal{I}_c \subset \mathbb{Z}$  the *cell (triangle) index set* and  $\mathcal{I}_e \subset \mathcal{I}_c \times \mathcal{I}_c$  the *edge index set*, which is assumed to be *symmetric*

$$(i, j) \in \mathcal{I}_e \Rightarrow (j, i) \in \mathcal{I}_e$$

and *irreflexive*

$$\forall i \in \mathbb{Z} : (i, i) \notin \mathcal{I}_e.$$

Each edge is described by a symmetric pair of edge indices  $((i, j), (j, i)) \in \mathcal{I}_e \times \mathcal{I}_e$ . Since we consider triangle meshes, each cell contributes to exactly three edges

$$\forall i \in \mathcal{I}_c : |\{j \in \mathcal{I}_c : (i, j) \in \mathcal{I}_e\}| = 3.$$

Furthermore,  $\bar{C} = (\bar{C}_i)_{i \in \mathcal{I}_v}$  denotes the vertex positions where  $\bar{C}_i \in \mathbb{R}^3$ . The *valence* of a vertex denotes the number of vertices that share an edge with this vertex denoted here by  $N$ . Then, we call a vertex *ordinary* if this vertex has valence six and consequently *extraordinary (EV)* if the vertex has a valence unequal to six. By  $\mathcal{N}_1(i)$  we denote the *1-ring of a vertex  $i$* , i.e. the set of all vertices that share an edge with this vertex and the vertex itself, i.e.  $|\mathcal{N}_1(i)| = N + 1$ . The definition of a  *$k$ -ring of a vertex  $i$*  denoted by  $\mathcal{N}_k(i)$  is then analog. Furthermore, the *1-ring of an element* denotes all triangles that share a vertex with this triangle and the triangle itself which is also denoted as *patch*. Then, we say that a patch is *regular* if the three inner vertices are ordinary and *irregular* if there is at least one inner extraordinary vertex (see Figure 2.5 for a regular and irregular patch).

As mentioned earlier subdivision defines smooth surfaces by applying a refinement process to an initial control polyhedral of arbitrary topology. Assume  $\mathcal{T}_0$  denotes a triangle mesh which contains only ordinary vertices denoted by  $\bar{C}^0$ . Then, for each vertex  $i \in \mathcal{I}_v$  we denote by  $\bar{C}_{\text{local}}^0 = (\bar{C}_0^0, \bar{C}_1^0, \dots, \bar{C}_6^0)$  all vertex positions in  $\mathcal{N}_1(i)$  where  $\bar{C}_0^0$  corresponds to the center vertex. From the theory of box splines we know that we can compute a refined mesh  $\mathcal{T}_1$  by quadrisecting each triangle of  $\mathcal{T}_0$  and then recompute the old vertex positions (called *vertex rule*) by

$$\bar{C}_0^1 = \frac{5}{8} \bar{C}_0^0 + \sum_{i=1}^6 \frac{1}{16} \bar{C}_i^0$$

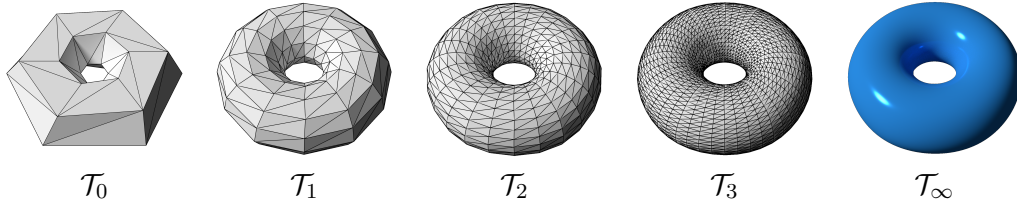


Figure 2.2.: Loop Subdivision surface: We depict the control mesh  $\mathcal{T}_0$ , the first three subdivision levels surface  $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$  and the resulting limit surface  $\mathcal{T}_\infty$ .

and the vertex positions for newly introduced vertices (called *edge rule*) by

$$\bar{C}_i^1 = \frac{\bar{C}_i^0 + 3\bar{C}_{i-1}^k + \bar{C}_i^k + 3\bar{C}_{i+1}^k}{8}, \quad i = 1, \dots, N, \quad (2.1)$$

where the index  $i$  has to be understood in modulo arithmetic. This process can easily be applied for some refinement step  $k$  which creates a triangle mesh  $\mathcal{T}_k$  with vertex position  $\bar{C}^k$ . In the limit, i.e.  $k \rightarrow \infty$ , this process produces a smooth surface. Here, the described weights correspond to the *quartic box spline* which produces smooth  $C^2$ -surfaces (again see e.g. [40]). Unfortunately, the only surface which can be constructed consisting only of ordinary vertices is topologically a torus (Figure 2.2).

In 1985, Charles Loop generalized the subdivision vertex rule of quartic triangular box splines to meshes of arbitrary topology, see Figure 2.1. Then, for each vertex  $i \in \mathcal{I}_v$  with valence  $N$  we compute the new vertex position at refinement level  $k$  by

$$\bar{C}_0^{k+1} = (1 - N\beta(N))\bar{C}_0^k + \sum_{i=1}^N \beta(N)\bar{C}_i^k \quad (2.2)$$

where  $\bar{C}_{\text{local}}^0 = (\bar{C}_0^k, \bar{C}_1^k, \dots, \bar{C}_N^k)$  follows analogously to the above definition and  $\beta(N) = \frac{1}{N} \left( \frac{5}{8} - \left( \frac{3}{8} + \frac{1}{4} \cos \left( \frac{2\pi}{N} \right) \right)^2 \right)$  denotes the generalized subdivision weight. A *subdivision surface* is then defined as the limit of this process, i.e.  $k \rightarrow \infty$ , where we denote the smooth surface by  $\mathcal{M} = \mathcal{T}_\infty$ , and we denote  $\mathcal{T}_0$  as the *coarse control mesh*. Let us remark that the subdivision process only inserts ordinary vertices, i.e. vertices with valence six. This implies that each subdivision of an irregular patch produces three regular and one irregular patch. As a consequence, the refined control meshes consist of large regions which are entirely regular (i.e. consist of quartic box spline patches) with isolated vertices whose valence is different from six. Let us remark that this process also applies to vectors of scalars instead of vectors of three-dimensional vectors  $\bar{C}_{\text{local}}$ .

Since (2.2) and (2.1) are linear combinations of  $\bar{C}_{\text{local}}^k$  we can express this subdivision scheme between level  $k$  and  $k+1$ , i.e.  $\bar{C}_{\text{local}}^k$  and  $\bar{C}_{\text{local}}^{k+1}$ , by the so-called the *local subdivision matrix*  $S \in \mathbb{R}^{N+1 \times N+1}$ , i.e.

$$\bar{C}_{\text{local}}^{k+1} = S\bar{C}_{\text{local}}^k. \quad (2.3)$$

Since Loop subdivision is a *stationary subdivision scheme*, i.e. the subdivision rules are the same for each subdivision step, we can study the *limit point*  $\bar{C}_i^\infty$  of  $\bar{C}_i^k$  for all  $i \in \mathcal{I}_v$  by examining the local subdivision process

$$\bar{C}_{\text{local}}^\infty = \lim_{k \rightarrow \infty} S^k \bar{C}_{\text{local}}^0. \quad (2.4)$$

**Eigenvalues and eigenvectors of the subdivision matrix** Now, we want to understand the limit process of (2.3) better. Therefore, we consider the *left*  $W^j$  and *right*  $V^j$  eigenvectors of  $S$ , i.e. for  $i = j, \dots, N + 1$

$$SV^j = \lambda_j V^j$$

and

$$W^j \cdot S = \lambda_j W^j$$

with *eigenvalues*  $\lambda_j$  ordered by magnitude, i.e.

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq |\lambda_4| \geq \dots \geq |\lambda_{N+1}|$$

and  $W^j \cdot V^i = \delta_{i,j}$  for  $i, j = 1, \dots, N + 1$ . From [131] we know that these eigenvalues are real, i.e.  $\lambda_j \in \mathbb{R}$  and the eigenvectors  $W^j$  and  $V^j$  can also be constructed to be real-valued. Then, we can consider the eigendecomposition of  $\bar{C}_{\text{local}}^0$ , i.e.

$$\bar{C}_{\text{local}}^0 = \sum_{j=0}^N d_j V^j,$$

where  $d_j = W^j \cdot \bar{C}_{\text{local}}^0$ . If we apply this decomposition to (2.4) we obtain

$$\bar{C}_{\text{local}}^\infty = \lim_{k \rightarrow \infty} S^k \sum_{j=1}^{N+1} d_j V^j = \lim_{k \rightarrow \infty} \sum_{j=1}^{N+1} \lambda_j^k d_j V^j. \quad (2.5)$$

From this it immediately follows that the limit surface cannot exist if  $\lambda_1 > 1$ . In particular, we have  $\lambda_1 = 1$  denoted as *dominant* eigenvalue,  $V^1 = (1, \dots, 1)^T$  and  $W^1 = (1 - N\ell(N), \ell(N), \dots, \ell(N))^T$  with  $\ell(N) = (\frac{3}{8\beta(N)} + N)^{-1}$ . Furthermore,  $S$  has the so-called *sub-dominant* eigenvalue

$$1 > \lambda_2 = \lambda_3 = \lambda = \frac{3}{8} + \frac{1}{4} \cos\left(\frac{2\pi}{N}\right)$$

of algebraic and geometric multiplicity 2 and

$$\lambda > \lambda_4 = \mu = \begin{cases} \frac{1}{8} & N = 3, \\ \lambda^2 & N = 4, 5, \\ \frac{1}{4} & N = 6, \\ \frac{3}{8} + \frac{1}{4} \cos\left(\frac{4\pi}{N}\right) & N > 6. \end{cases}$$

Using this information we immediately obtain from (2.5)

$$\bar{C}_{\text{local}}^\infty = d_1 V^1 \quad \text{with } d_1 = W^1 \cdot \bar{C}_{\text{local}}^0,$$

i.e. we have a well-defined limit point

$$\bar{C}_i^\infty = (1 - N\ell(N))\bar{C}_i^0 + \sum_{j \in \mathcal{N}_1(i)} \ell(N)C_j^0 \quad (2.6)$$

for all extraordinary vertices  $i \in \mathcal{I}_v$ , such that we can compute the limit position of an arbitrary control vertex without going to the limit.

**Domain manifold** The domain of a Loop subdivision surface is a topological manifold obtained by gluing together copies of a standard triangle. First, we define the *domain pre-manifold*

$$\hat{\Omega} = \Delta \times \mathcal{I}_c \quad (2.7)$$

as the cartesian product of the unit triangle

$$\Delta = \{(\xi_1, \xi_2) \in \mathbb{R}^2 \mid 0 \leq \xi_1 + \xi_2 \leq 1, \xi_1, \xi_2 \geq 0\} \subset \mathbb{R}^2 \quad (2.8)$$

and the cell index set  $\mathcal{I}_c$ . To construct the domain manifold from the pre-manifold we need to identify points on the common boundary of two triangles. Therefore, we associate with each edge index  $(i, j) \in \mathcal{I}_e$  a displacement  $\delta_j^i$ , which is one of the 18 isometries ( $3 \cdot 3 \cdot 2$ ) that map the unit triangle to one of its three neighbor triangles obtained by reflection and index permutation. These displacements in particular identify the common edges of neighboring triangles and their orientation with respect to the two triangles. They need to satisfy the following two conditions. Firstly, the identification of the common edge and its orientation has to be consistent for each edge, thus

$$\forall (i, j) \in \mathcal{I}_e : \delta_j^i = (\delta_i^j)^{-1}.$$

Secondly, each edge of a triangle is identified with exactly one edge of another triangle,

$$\forall (i, j) \in \mathcal{I}_e : \forall (i, k) \in \mathcal{I}_e : \delta_j^i(\Delta) \cap \Delta = \delta_k^i(\Delta) \cap \Delta \Rightarrow j = k.$$

Two points of the pre-manifold  $(\xi, i), (\eta, j) \in \hat{\Omega}$  are identified, denoted by  $(\xi, i) \sim (\eta, j)$ , if  $(i, j)$  is an edge index and the displacement  $\delta_j^i$  transforms  $\eta$  into  $\xi$ . This implies that  $\xi$  and  $\eta$  are located on the boundary of the unit triangle. We denote with  $\sim$  the reflexive and transitive closure of this relation. This closure leads to the obvious identification of common vertices. Then, the topological manifold

$$\Omega = (\Delta \times \mathcal{I}_c) / \sim \quad (2.9)$$

is the *domain manifold* of the Loop subdivision surface.

**Subdivision basis functions** Now, we describe the construction of basis functions needed for finite element simulation.

Therefore, we consider a triangulation  $\mathcal{T}_k$  and the spaces

$$L^k = \{f \in C^0(\mathcal{T}_k) \mid f|_i \in \mathcal{P} \forall i \in \mathcal{I}_c\} \quad (2.10)$$

of piecewise linear functions on  $\mathcal{T}_k$ . Here, each function is uniquely described by its *nodal values*, i.e., by its values at the vertices of the triangulation. Analog to the local subdivision matrix we can define a

## 2. Foundations

Loop subdivision operator  $S_k : L^{k-1} \rightarrow L^k$  that transform a piecewise linear function  $f^{k-1} \in L^{k-1}$  into a function  $f^k = S_k f^{k-1}$ . Again,  $S_k$  reflects the subdivision vertex rule (2.2) and edge rule (2.1). The space of *Loop subdivision splines* consists of the limit functions generated by these subdivision operators,

$$\left\{ \lim_{k \rightarrow \infty} S_k S_{k-1} \cdots S_1 f : f \in L^0 \right\}.$$

Now we have everything at hand to define the *nodal functions* for all  $i \in \mathcal{I}_v$  by

$$\Phi_i = \lim_{k \rightarrow \infty} S_k S_{k-1} \cdots S_1 \Lambda_i,$$

where  $\Lambda_i \in L^0$  denotes the associated piecewise linear hat function which takes the nodal value 1 at the associated vertex  $i \in \mathcal{I}_v$  and 0 else. Assume that all extraordinary vertices of  $\mathcal{T}_0$  are isolated, i.e. two extraordinary vertices are separated by at least one ordinary vertex.

**Theorem 2.2.1.** The set of Loop subdivision nodal functions are globally linearly independent over the domain manifold  $\Omega$ , i.e. if  $\forall (\xi, k) \in \Omega : \sum_{i \in \mathcal{I}_v} a_i \Phi_i(\xi, k) = 0$ , then  $a_i = 0$  for all  $i \in \mathcal{I}_v$ .

*Proof.* See Corollary 2.3 in [112]. □

Because of the compact support of the subdivision rules, the support of a Loop basis function is the two-ring of the vertex  $i \in \mathcal{I}_v$  in the coarsest triangulation  $\mathcal{T}_0$ . If the support does not contain EVs in its interior, then it consists of 24 triangles at the coarsest level and the Loop basis function is the  $C^2$ -smooth quartic box spline. Now we have everything at hand to introduce a *Loop subdivision surface*

$$X : \Omega \rightarrow \mathbb{R}^3 : X(\xi, k) = \sum_{i \in \mathcal{I}_v} \bar{C}_i \Phi_i(\xi, k), \quad (2.11)$$

which is obtained by assigning control points  $\bar{C} = (\bar{C}_i)_{i \in \mathcal{I}_v}$ , where  $\bar{C}_i \in \mathbb{R}^3$ , to the Loop basis functions with  $\xi = (\xi_1, \xi_2) \in \Delta$ . For a visualization of the curved triangles  $X(\Delta, k)$  see Figure 2.3.

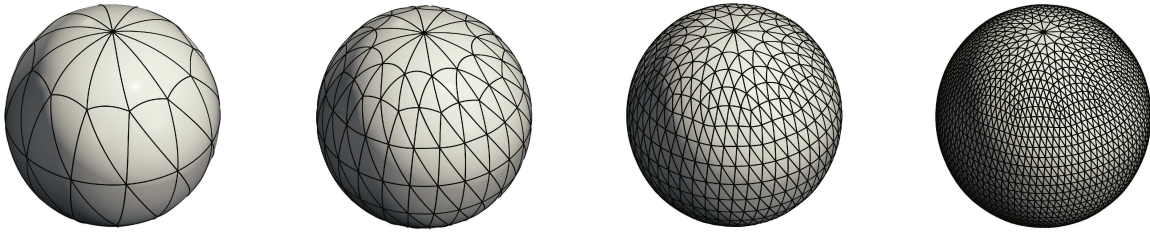


Figure 2.3.: Visualization of a spherical subdivision limit surface with extraordinary vertices of valence 5 and 12. Instead of the control meshes we depicted the limit surface with the smooth curved triangles on different subdivision levels.

**Eigenfunctions, the characteristic map and spline rings** Now, we have almost everything at hand to understand the behavior in the vicinity of an extraordinary vertex. Therefore, we consider the extended left eigenvectors  $W^j \in \mathbb{R}^{N+6}$  which correspond to all vertices in the 1-ring of an irregular element (still ordered with respect to magnitude of the corresponding eigenvalues  $\lambda_j \in \mathbb{R}$ ,

$j = 1, \dots, N + 6$ ). Then, we define for fixed  $k$  the *eigenfunctions*  $\forall j = 1, \dots, N + 6$  by

$$\nu_j(\xi, k) = \sum_{i=1}^{N+6} W_i^j \Phi_i(\xi, k), \quad \xi \in \Delta. \quad (2.12)$$

Here, the term eigenfunction is reflected by the following scaling relation (holds for  $N > 3$ )

$$\nu_j(\xi/2^l, k) = \lambda_j^l \nu_j(\xi, k) \quad \forall \xi \in \Delta, l > 0. \quad (2.13)$$

Then, we can decompose the unit triangle  $\Delta$  by

$$\Delta = \bigcup_{l=1}^{\infty} \Delta_l, \quad \text{where } \Delta_1 = \Delta \setminus \frac{1}{2}\Delta \text{ and } \Delta_{l+1} = \frac{1}{2}\Delta_l.$$

This means that  $\nu_j(\Delta_{l+1}, k) = \lambda_j^l \nu_j(\Delta_1, k)$ , i.e.  $\nu_j(\Delta_{l+1}, k)$  is a scaled copy of  $\nu_j(\Delta_1, k)$ . For  $N = 3$  the situation changes barely. Here, the subdivision matrix cannot be diagonalized, but a Jordan decomposition can be computed. This contains only one non-trivial Jordan block and an efficient factorization can be used (see [131]). Then, the definitions apply again as above. Now we can define the *characteristic map* by

$$\Psi(\xi, k) = \begin{pmatrix} \nu_2(\xi, k) \\ \nu_3(\xi, k) \end{pmatrix}. \quad (2.14)$$

Then, we have the scaling relation  $\Psi(\Delta_{l+1}, k) = \lambda^l \Psi(\Delta_1, k)$  for all  $k \in \mathcal{I}_c$  in the neighborhood of an extraordinary vertex, and, since both eigenfunctions correspond to the same eigenvalue, the sub-dominant eigenvalue  $\lambda$ . The fact that  $\nu_j(\Delta_l, k)$  is again a quartic box spline for all  $k \in \mathcal{I}_c$ ,  $l > 1$  and  $j \in \mathcal{I}_v$  motivated the term *spline ring* and is the crucial difference between multi-patched box spline surfaces (including NURBS) and subdivision surfaces. In subdivision, surfaces of arbitrary topology are obtained by solving the  $N$ -sided hole-filling problem instead of gluing together patches. The extraordinary vertices are at the center of irregular patches which can be thought of as consisting of an infinite geometric sequence of rings of regular patches (see Figure 2.4). From the scaling relation we see that the size of the sequence of rings is determined by the so-called *sub-dominant eigenvalue*  $\lambda$  which depends on the valence of the extraordinary vertex. Because of these observations the analytic properties of the limit surface are given by the properties of the quartic box spline, except at the extraordinary vertex where we have a singularity.

**Smoothness of Loop's subdivision surfaces** Finally, we have everything almost everything at hand to list the smoothness properties of Loop's subdivision surfaces. First, let us remark that from the scaling relation (2.13) we see for  $j = 2, 3$  and  $N > 6$

$$\frac{\partial \nu_j(\xi/2^l, k)}{\partial \xi} = (2\lambda)^l \frac{\partial \nu_j(\xi, k)}{\partial \xi} \quad \forall \xi \in \Delta, k \in \mathcal{I}_c,$$

i.e. the derivatives of  $\nu_j$  diverge because  $\lambda > \frac{1}{2}$  which means that the first derivatives of Loop's subdivision surface  $X$  diverge in this setting. Let us now come to the breakthrough result by Reif [116] which characterizes a  $C^1$ -smooth subdivision surface.

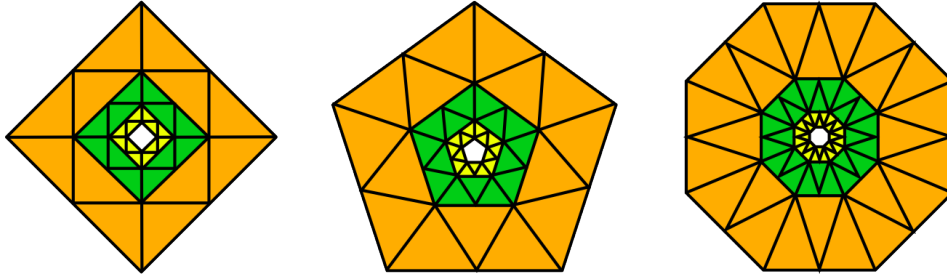


Figure 2.4.:  $N$ -sided hole filling problem for  $N = 4, 5, 8$ . Here, the green ring is always a  $\lambda$ -scaled copy and the yellow ring a  $\lambda^2$ -scaled copy of the orange ring, where  $\lambda$  denotes the subdominant eigenvalue of the subdivision scheme. For subdivision surfaces, the rings are spline rings and the infinite sequence of these rings is the approach of subdivision to construct smooth surfaces of arbitrary topology with a singularity at the centered extraordinary vertex.

**Theorem 2.2.2.** If  $\lambda_2 = \lambda_3$ ,  $1 > |\lambda_2| > |\lambda_4|$ , is a real eigenvalue with algebraic and geometric multiplicity 2 and if the characteristic map is regular (i.e.  $\frac{\partial \Psi(\xi, k)}{\partial \xi} \neq 0$  for all  $\xi \in \Delta$  and  $k \in \mathcal{I}_c$ ) and injective, then subdivision surface

$$\tilde{X} = X \circ \Psi^{-1} : \tilde{\Omega} \mapsto \mathbb{R}^3$$

is  $C^1$  at an extraordinary vertex where

$$\tilde{\Omega} = \Psi(\Omega) = \sum_k \Psi(\Delta, k)$$

denotes the *embedded domain manifold*.

*Proof.* See Theorem 3.6 in [116]. □

Applying this fundamental result to Loop's subdivision scheme, Umlauf [137] showed the following result.

**Theorem 2.2.3.** Loop subdivision surfaces are  $C^1$ -smooth using the characteristic map embedding.

*Proof.* See Lemma 6.1 in [137]. □

Unfortunately, for finite element or isogeometric simulations we don't want to compute the inverse of the characteristic map and instead work with  $X$ . Here, Reif and Schröder [117] underlined the justification for Loop subdivision surfaces to be used in higher order finite element simulations.

**Theorem 2.2.4.** Loop's subdivision surfaces are  $H^2$ -integrable, i.e.  $\|X\|_{H^2} < \infty$ .

In particular,

$$X \in \begin{cases} C^2 & \text{iff all vertices have valence 6,} \\ W^{2,p}, p < \infty & \text{iff all vertices have valence 4, 5 or 6,} \\ W^{2,p}, p < \frac{2 \ln |\lambda|}{2 \ln |\lambda| - \ln |\mu|} & \text{else.} \end{cases}$$



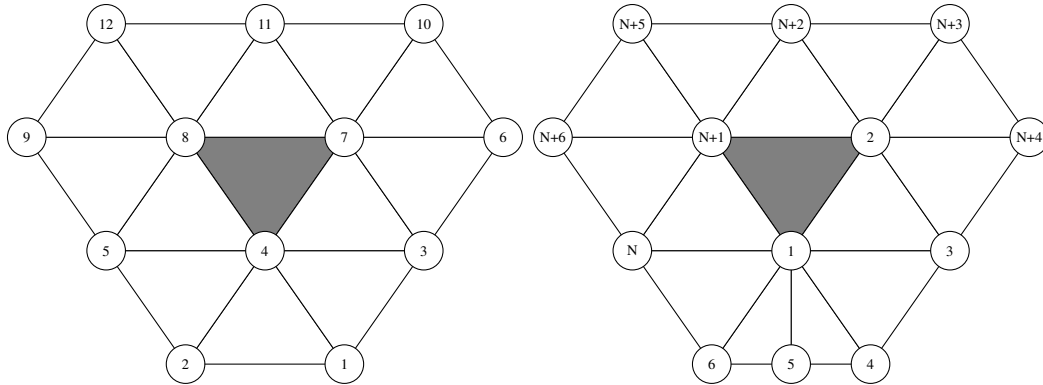


Figure 2.5.: Local numbering of a regular box spline patch with 12 control points on the right and the local numbering of an irregular patch with  $N + 6$  nodes on the left.

*Proof.* See Theorem 2 in [117]. □

**The natural parameterization** Let us now describe the evaluation algorithm by Stam [131] in detail. First we assume  $X(\xi, k)$  is the left gray patch in Figure 2.5 with  $(\xi, k) \in \Omega$ . Because all vertices are ordinary we can write

$$X(\xi, k) = \sum_{i=1}^{12} \bar{C}_i N_i(\xi_1, \xi_2) = \bar{\mathbf{C}}^T \mathbf{N}(v, w),$$

where  $\bar{C}_i \in \mathbb{R}^d$  ( $d = 1, 2, 3$ ) are the coordinates of the nodes with the local numbering as in Figure 2.5 and  $N_i$  are the corresponding shape functions (see Appendix A.1).

The situation around an extraordinary vertex of valence  $N$  is different. In this case we have  $K = N + 6$  vertices surrounding the patch. We assume that the extraordinary vertex is located at  $\xi = (0, 0)$ , i.e. at the origin of the unit triangle. Then we can write the control vertices in a  $K \times 3$ -matrix given by

$$\mathbf{C}_0 = (\bar{C}_1^0, \bar{C}_1^0, \dots, \bar{C}_K^0).$$

Then we define the extended  $K \times K$  subdivision matrix which describes the next subdivision step entirely by

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix}$$

and an even bigger subdivision matrix by

$$\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \\ \mathbf{A}_{31} & \mathbf{A}_{32} \end{pmatrix}$$

## 2. Foundations

---

and we get the new vertices by

$$\mathbf{C}_1 = \mathbf{A}\mathbf{C}_0 \quad \text{and} \quad \tilde{\mathbf{C}}_1 = \tilde{\mathbf{A}}\mathbf{C}_0,$$

where  $\tilde{\mathbf{C}}_1 = (\tilde{C}_1^1, \tilde{C}_2^1, \dots, \tilde{C}_K^1, \tilde{C}_{K+1}^1, \dots, \tilde{C}_M^1)$  with  $M = K + 6 = N + 12$ . In the definition of  $\mathbf{A}$  and  $\tilde{\mathbf{A}}$ , we used

$$\mathbf{A}_{11} = \begin{pmatrix} 1 - N\beta(N) & \beta(N) & \beta(N) & \beta(N) & \beta(N) & \dots & \beta(N) & \beta(N) & \beta(N) \\ c & c & d & 0 & 0 & \dots & 0 & 0 & d \\ c & d & c & d & 0 & \dots & 0 & 0 & 0 \\ c & 0 & d & c & d & \dots & 0 & 0 & 0 \\ & & & & & \vdots & & & \\ c & d & 0 & 0 & 0 & \dots & 0 & d & c \end{pmatrix},$$

$$\mathbf{A}_{21} = \frac{1}{16} \begin{pmatrix} 2 & 6 & 0 & 0 \dots & 0 & 0 & 6 \\ 1 & 10 & 1 & 0 \dots & 0 & 0 & 1 \\ 2 & 6 & 6 & 0 \dots & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \dots & 0 & 1 & 10 \\ 2 & 0 & 0 & 0 \dots & 0 & 6 & 6 \end{pmatrix}, \quad \mathbf{A}_{22} = \frac{1}{16} \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix},$$

$$\mathbf{A}_{31} = \frac{1}{8} \begin{pmatrix} 0 & 3 & 0 & 0 \dots & 0 & 0 & 1 \\ 0 & 3 & 0 & 0 \dots & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \dots & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \dots & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \dots & 0 & 1 & 3 \end{pmatrix} \quad \text{and} \quad \mathbf{A}_{32} = \frac{1}{8} \begin{pmatrix} 3 & 1 & 0 & 0 & 0 \\ 1 & 3 & 1 & 0 & 0 \\ 0 & 1 & 3 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 & 3 \end{pmatrix}.$$

where  $c = \frac{3}{8}$  and  $d = \frac{1}{8}$ . Then, by repeating this process, we obtain an infinite sequence of control vertices by

$$\tilde{\mathbf{C}}_n = \tilde{\mathbf{A}}\mathbf{C}_{n-1} = \tilde{\mathbf{A}}\mathbf{A}^{n-1}\mathbf{C}_0, \quad n \geq 1.$$

For each  $n \geq 1$  the subdivision process produces 3 regular patches and 1 irregular patch (see Figure 2.6). The vertices are now in  $\tilde{\mathbf{C}}_n$  and we only have to pick the right vertices corresponding to the desired patch. Therefore we define the picking matrices  $\mathbf{P}_m$  for  $m = 1, 2, 3$ , such that we choose the right vertices for three regular patches. Hence, we define

$$\tilde{\mathbf{C}}_{n,m} = \mathbf{P}_m \tilde{\mathbf{C}}_n, \quad m = 1, 2, 3.$$

Each row of the picking matrix  $\mathbf{P}_m$  is filled with zeros except for the column corresponding to the index which is filled with a one (see Figure 2.6). Each surface patch is then defined as follows:

$$X_{n,m}(\xi, k) = \tilde{\mathbf{C}}_{n,m}^T \mathbf{N}(\xi_1, \xi_2) = \mathbf{P}_m^T \tilde{\mathbf{C}}_n^T \mathbf{N}(\xi_1, \xi_2).$$

Since this is just a parameterization of a part of our triangle, we seek for a parameterization of the whole

triangle  $X(\xi, k)$ . As explained above we have an infinite number of patches  $\Delta_m^n$  which are given by

$$\begin{aligned}\Delta_1^n &= \{(\xi_1, \xi_2) \mid \xi_1 \in [2^{-n}, 2^{-n+1}] \text{ and } \xi_2 \in [0, 2^{-n+1} - \xi_1]\} \\ \Delta_2^n &= \{(\xi_1, \xi_2) \mid \xi_1 \in [0, 2^{-n}] \text{ and } \xi_2 \in [0, \xi_1]\} \\ \Delta_3^n &= \{(\xi_1, \xi_2) \mid \xi_1 \in [0, 2^{-n}] \text{ and } \xi_2 \in [2^{-n}, 2^{-n+1} - \xi_1]\}.\end{aligned}$$

The surface patch is then defined by its restriction to each of these triangle

$$X(\xi, k) |_{\Delta_m^n} = X_{n,m}(t_{n,m}(\xi, k)) = \mathbf{C}_0^T (\mathbf{P}_m \bar{\mathbf{A}} \mathbf{A}^{n-1})^T \mathbf{N}(t_{n,m}(\xi_1, \xi_2))$$

with

$$\begin{aligned}t_{n,1}(\xi_1, \xi_2) &= (2^n \xi_1 - 1, 2^n \xi_2), \\ t_{n,2}(\xi_1, \xi_2) &= (1 - 2^n \xi_1, 1 - 2^n \xi_2), \\ t_{n,3}(\xi_1, \xi_2) &= (2^n \xi_1, 2^n \xi_2 - 1).\end{aligned}$$

Furthermore the picking matrices are

$$\begin{aligned}P_1(N) &= (\mathbb{I}_2 \ \mathbb{I}_{N+3} \ \mathbb{I}_{N+4} \ \mathbb{I}_3 \ \mathbb{I}_1 \ \mathbb{I}_{N+1} \ \mathbb{I}_{N+2} \ \mathbb{I}_{N+7} \ \mathbb{I}_{N+8} \ \mathbb{I}_{N+9} \ \mathbb{I}_{N+10} \ \mathbb{I}_{N+5})^T, \\ P_2(N) &= (\mathbb{I}_{N+2} \ \mathbb{I}_{N+1} \ \mathbb{I}_{N+5} \ \mathbb{I}_{N+10} \ \mathbb{I}_{N+7} \ \mathbb{I}_{N+3} \ \mathbb{I}_2 \ \mathbb{I}_1 \ \mathbb{I}_N \ \mathbb{I}_{N+6} \ \mathbb{I}_3 \ \mathbb{I}_{N+4})^T, \\ P_3(N) &= (\mathbb{I}_{N+1} \ \mathbb{I}_{N+2} \ \mathbb{I}_2 \ \mathbb{I}_1 \ \mathbb{I}_N \ \mathbb{I}_{N+6} \ \mathbb{I}_{N+5} \ \mathbb{I}_{N+10} \ \mathbb{I}_{N+7} \ \mathbb{I}_{N+3} \ \mathbb{I}_{N+11} \ \mathbb{I}_{N+12})^T,\end{aligned}$$

where  $\mathbb{I}_j \in \mathbb{R}^{N+12}$  denote the zero vectors except at the  $j$ -th position where the value is one.

Since we will evaluate the basis functions only on a given set of quadrature points, this can be done in a pre-processing step. For a more efficient implementation using the eigenstructure of  $\mathbf{A}$  see the original paper [131] (where the computation is made almost independent of  $n$ ). Here, the extended subdivision matrix  $\mathbf{A}$  can be diagonalized for  $N > 3$ , i.e.

$$A = V \Lambda V^{-1}.$$

Hence, we can write

$$X_{n,m}(t_{n,m}(\xi, k)) = (V^{-1} \mathbf{C}_0)^T \Lambda^{n-1} (\mathbf{P}_m \tilde{\mathbf{A}} V)^t \mathbf{N}(t_{n,m}(\xi_1, \xi_2)).$$

For  $N = 3$  the extended subdivision matrix has a non-trivial Jordan block, but can still be treated in a similiar fashion (see [131]). For a detailed description of  $V$ ,  $J$ , and  $\Lambda$  see [131], where the eigendecomposition  $A = V \Lambda V^{-1}$  is not ordered with respect to the magnitude of the eigenvalues  $\lambda_j$ . Finally, let us remark that the eigenfunctions  $\nu_i$  defined before are given by

$$\nu_i(\xi, k) |_{\Delta_m^n} = (\Lambda^{n-1} (\mathbf{P}_m \tilde{\mathbf{A}} V)^t \mathbf{N}(t_{n,m}(\xi_1, \xi_2)))_i,$$

if  $V$  is ordered with respect to the magnitude of the eigenvalues. Hence, we could evaluate the characteristic map  $\Psi(\xi, k) = (\nu_2(\xi, k), \nu_3(\xi, k))^T$  as well.

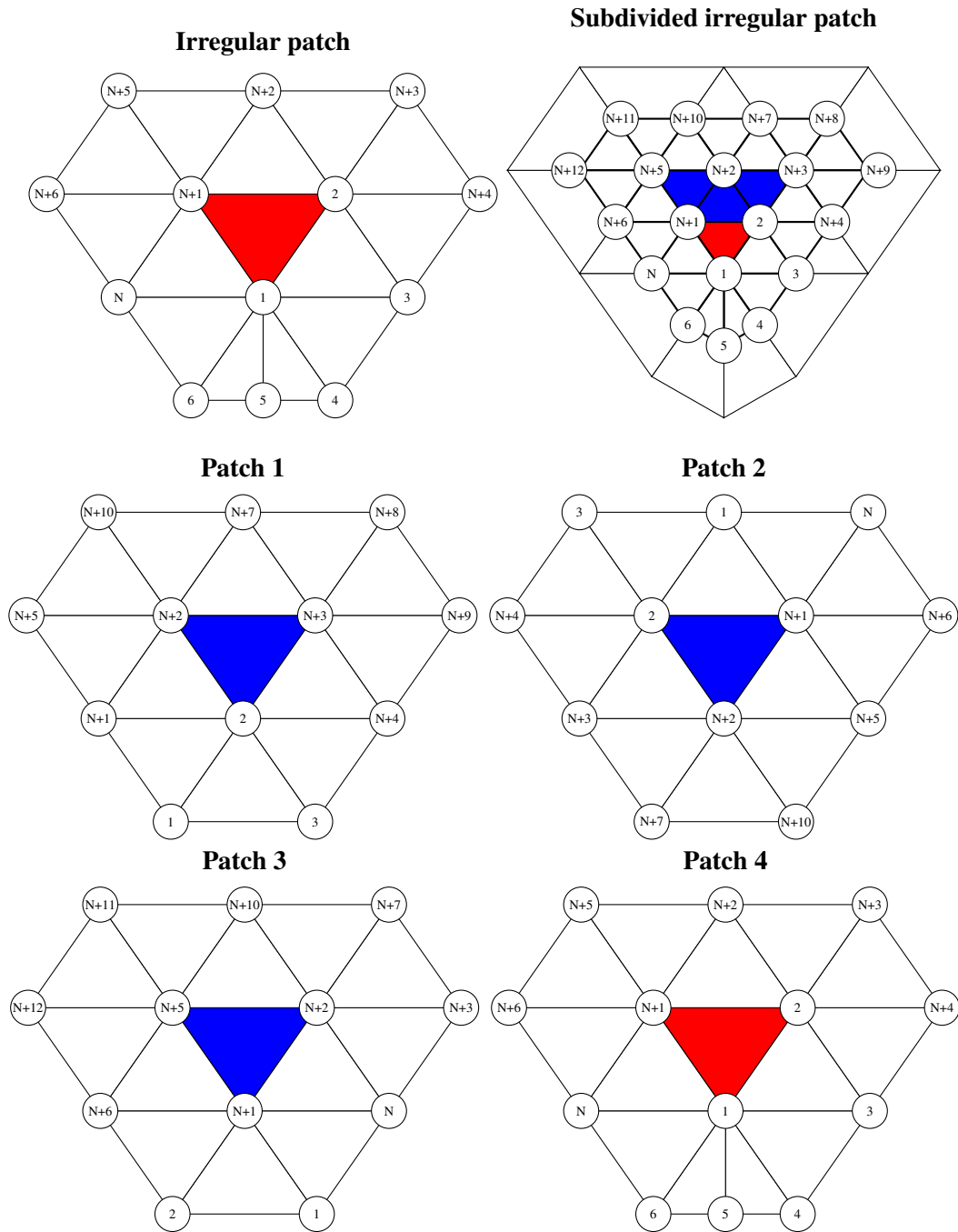


Figure 2.6.: Patch refinement near an extraordinary vertex. On the top the corresponding irregular patch is filled red on the left. On the right after one subdivision step the three blue triangles are again regular and the red patch is still irregular. On the bottom all four newly created patches.





## 3. Isogeometric Subdivision Method

In this chapter we focus on the formulation and implementation of subdivision surfaces in the context of numerical simulation using the isogeometric approach. A particular focus of this chapter is on the *convergence order* of Loop's subdivision for isogeometric simulations, which will be analyzed through a detailed *experimental convergence study* in dependence of an efficient assembly of the corresponding isogeometric matrices. Let us remark that Loop's subdivision surfaces were first considered for finite element simulations by Cirak, Ortiz and Schröder [33] before the invention of the isogeometric paradigm and Cirak and co-authors [32] suggested to use subdivision surfaces as integrated tool for modeling and simulation.

First we transform the concepts from differential geometry to our discrete setting. Because the basis functions of subdivision surface are almost everywhere  $C^2$ -smooth, a point-wise definition of all concepts from Chapter 2 is straightforward. Furthermore, we will focus on the efficient numerical integration based on quadrature. Therefore, we consider three model problems, the *Laplace-Beltrami* equation

$$-\Delta_{\mathcal{M}}u = f \text{ on } \mathcal{M}, \quad (3.1)$$

the *surface bi-Laplacian* equation

$$(-\Delta_{\mathcal{M}})^2u = f \text{ on } \mathcal{M} \quad (3.2)$$

as well as the eigenvalue problem

$$-\Delta_{\mathcal{M}}u = \lambda u \text{ on } \mathcal{M}. \quad (3.3)$$

We will consider the assembly of the corresponding isogeometric matrices for the model problems in detail and discuss different quadrature schemes. In addition we discuss the previously proposed adaptive strategies around extraordinary vertices [104] and provide a look-up table for the mid-edge quadrature based on [70] which facilitates the implementation and results in a fairly robust simulation tool and a very efficient assembly of finite element matrices.

### 3.1. Spatial Discretization

In the previous chapter we have seen that Loop subdivision surfaces have a basis function representation cf. (2.11). We will now introduce the notions for the differential geometric concepts in our discrete setting. Therefore, let us repeat the definition of a Loop subdivision from a domain manifold  $\Omega$  to the embedded surface  $\mathcal{M}$  by

$$X : \Omega \rightarrow \mathbb{R}^3 : X(\xi, k) = \sum_{i \in \mathcal{I}_v} \bar{C}_i \Phi_i(\xi, k),$$

### 3. Isogeometric Subdivision Method

---

where  $k \in \mathcal{I}_c$  denotes an element index,  $\xi \in \Delta$  and  $\bar{C} = (\bar{C}_i)_{i \in \mathcal{I}_v}$  denotes the vector of three-dimensional control point coordinates  $\bar{C}_i$ . In the context of isogeometric analysis, this parameterization  $X$  of the manifold is referred to as the *geometry mapping*. Then, the definition of the Jacobian and Hessian of  $X$  become

$$DX(\xi, k) = (X_{,1} \ X_{,2}) = \sum_{i \in \mathcal{I}_v} \bar{C}_i \begin{pmatrix} \Phi_{i,1}(\xi, k) & \Phi_{i,2}(\xi, k) \end{pmatrix} = \sum_{i \in \mathcal{I}_v} \bar{C}_i \nabla_{\xi} \Phi_i(\xi, k),$$

where  $\Phi_{i,j}(\xi, k) = \frac{\partial \Phi_i(\xi, k)}{\partial \xi_j}$  denotes the first partial derivative with respect to  $\xi_j$  for  $j = 1, 2$  and

$$D^2 X(\xi, k) = \begin{pmatrix} X_{,11} & X_{,12} \\ X_{,21} & X_{,22} \end{pmatrix} = \sum_{i \in \mathcal{I}_v} \bar{C}_i \begin{pmatrix} \Phi_{i,11}(\xi, k) & \Phi_{i,12}(\xi, k) \\ \Phi_{i,21}(\xi, k) & \Phi_{i,22}(\xi, k) \end{pmatrix} = \sum_{i \in \mathcal{I}_v} \bar{C}_i \nabla_{\xi}^2 \Phi_i(\xi, k),$$

where  $\Phi_{i,jl}(\xi, k) = \frac{\partial^2 \Phi_i(\xi, k)}{\partial \xi_j \partial \xi_l}$  denotes the second partial derivative with respect to  $\xi_j$  and to  $\xi_l$  for  $j, l = 1, 2$ .

With these three ingredients at hand we can now define the discrete counterparts to the geometric objects from Chapter 2. First the matrix of the first fundamental form is now denoted by

$$G(\xi, k) = DX(\xi, k)^T DX(\xi, k)$$

instead of  $g$ , the entries of the second fundamental form are

$$B_{ij}(\xi, k) = X_{,ij}(\xi, k) N(\xi, k)$$

instead of  $b$  where

$$N(\xi, k) = \frac{Y_{,1}(\xi, k) \times Y_{,2}(\xi, k)}{|Y_{,1}(\xi, k) \times Y_{,2}(\xi, k)|}$$

denotes the discrete normal to  $Y(\xi, k)$ . Second, the notion of the Weingarten maps  $\mathbf{s}_p$  and  $\mathbf{s}_{\xi}$  and the mean curvature  $\mathbf{h}$  retain. The evaluation for a point  $(\xi, k) \in \Omega$  becomes

$$\mathbf{s}_p(\xi, k) = DX(\xi, k) \cdot G^{-1}(\xi, k) \cdot B(\xi, k) \cdot G^{-1}(\xi, k) \cdot DX(\xi, k)^T$$

resp.

$$\mathbf{s}_{\xi}(\xi, k) = B(\xi, k) \cdot G^{-1}(\xi, k)$$

and  $\mathbf{h}(\xi, k) = \text{tr}(\mathbf{s}_p(\xi, k)) = \text{tr}(\mathbf{s}_{\xi}(\xi, k)) = \text{tr}(B(\xi, k) \cdot G^{-1}(\xi, k))$  where  $p = X(\xi, k) \in \mathcal{M}$ .

Then, the basis functions  $\varphi_i$  of the associated *isogeometric function space* are the push-forwards of the subdivision basis functions

$$\varphi_i(p) = \Phi_i \circ X^{-1}(p),$$

where  $p = X(\xi, k)$ , i.e. the basis functions are defined by the discretization  $X$ . We define the *discretization space* as

$$V_h = \left\{ u_h \in \text{span}_{i \in \mathcal{I}_v} \{ \varphi_i \} \right\}. \quad (3.4)$$



Here,  $h$  indicates the grid size of the control mesh, i.e.  $h = \max_{(i,j) \in \mathcal{I}_e} |C_i - C_j|$ . Furthermore, for a function  $u : \mathcal{M} \rightarrow \mathbb{R}$

$$\nabla_{\mathcal{M}} u(p) = (DX \cdot G^{-1} \nabla_{\xi}(u \circ X))(\xi, k)$$

is the (*embedded*) *tangential gradient* or *surface gradient* and

$$\Delta_{\mathcal{M}} u(p) = \frac{1}{\sqrt{\det G}} \operatorname{div}_{\xi} \left( \sqrt{\det G} G^{-1} \nabla_{\xi}(u \circ X) \right) (\xi, k)$$

the *Laplace-Beltrami operator*, where  $p = X(\xi, k)$ . Additionally if  $u$  has a basis function representation, i.e.  $u(p) = \sum_{i \in \mathcal{I}_v} u_i \varphi_i(p)$ , then

$$\nabla_{\mathcal{M}} u(p) = \sum_{i \in \mathcal{I}_v} u_i (DX \cdot G^{-1} \nabla_{\xi} \Phi_i) (\xi, k)$$

and

$$\Delta_{\mathcal{M}} u(p) = \sum_{i \in \mathcal{I}_v} u_i \frac{1}{\sqrt{\det G}} \operatorname{div}_{\xi} \left( \sqrt{\det G} G^{-1} \nabla_{\xi} \Phi_i \right) (\xi, k).$$

## 3.2. Weak Formulation

In the classical context of *calculus of variations* we try to find a solution of the *weak formulation* for a given partial differential equation instead of solving the *strong formulation* of the partial differential equation (e.g. problems (3.1) – (3.3)). Usually this is done by considering a suitable energy functional of which a weak form is derived by calculating the variation. Instead of considering the energy functional we directly start with the weak formulation. For the numerical treatment of partial differential equations using the *isogeometric subdivision approach* we make use of this weak formulation as it is done in the classical *finite element method*.

In the following we assume such a weak formulation is given. As general type we consider variational problems

$$a(u, v) = \ell_f(v) \quad \forall v \in V, \tag{3.5}$$

where  $a : V \times V \rightarrow \mathbb{R}$  is a bilinear form and  $\ell_f : V \rightarrow \mathbb{R}$  a linear form for a given function  $f \in L^2(\mathcal{M})$  on the function space  $V$ , where  $\ell_f(v) = \int_{\mathcal{M}} f v \, da$ . Furthermore we assume problem (3.5) has a unique solution. If  $a$  is symmetric, coercive and bounded on a Hilbert space  $V$ , i.e.

$$a(u, v) = a(v, u),$$

there exists a constant  $c > 0$  such that

$$a(u, u) \geq c \|u\|^2, \quad \forall u \in V,$$

and there exists a constant  $M \in \mathbb{R}$  such that

$$a(u, v) \leq M \|u\| \|v\|, \quad \forall u, v \in V,$$

### 3. Isogeometric Subdivision Method

---

where  $\|\cdot\|$  denotes the norm of  $V$ , then there exists a unique solution  $u \in V$  of the variational problem (3.5). For a proof see e.g. ([30, 22]).

For our test problems (3.1) – (3.3) the weak formulation is well known. The bilinear form  $a$  reads

$$a(u, v) = \int_{\mathcal{M}} \nabla_{\mathcal{M}} u \cdot \nabla_{\mathcal{M}} v \, da$$

and

$$a(u, v) = \int_{\mathcal{M}} \Delta_{\mathcal{M}} u \Delta_{\mathcal{M}} v \, da$$

for the Laplace-Beltrami (3.1) and the surface bi-Laplacian problem (3.2), respectively. The existence and uniqueness of the model problems can be found e.g. in [21, 49, 6]. Let us remark that on closed, smooth surfaces the kernel of the Laplace-Beltrami and the surface bi-Laplacian operators are the constant functions. Hence, we consider the weak function spaces

$$V = H^1(\mathcal{M}) \cap \left\{ \int_{\mathcal{M}} v \, da = 0 \right\}$$

for problem (3.1) and

$$V = H^2(\mathcal{M}) \cap \left\{ \int_{\mathcal{M}} v \, da = 0 \right\}$$

for problem (3.2).

Next, we want to approximate the solution of the weak problem (3.5) in the finite-dimensional subspaces  $V_h$ . The associated *Galerkin approximation* asks for the unique solution  $u_h \in V_h$  of the discrete variational problem

$$a(u_h, v_h) = \ell_f(v_h), \quad \forall v_h \in V_h, \quad (3.6)$$

i.e. instead of solving the weak formulation in  $V$  we solve the weak formulation in  $V_h$ . Due to the known  $H^2$ -regularity of the Loop subdivision splines (see Chapter 2 or [117]) this Galerkin approximation is *conforming*, i.e.  $V_h \subset V$ . Because  $\{\varphi_j\}_{j \in \mathcal{I}_v}$  is a basis of  $V_h$ , (3.6) transforms to

$$a(u_h, \varphi_j) = \ell_f(\varphi_j), \quad \forall j \in \mathcal{I}_v.$$

Using the basis expansion  $u_h(p) = \sum_{i \in \mathcal{I}_v} u_i \varphi_i(p)$  for  $u_h$  with coefficients  $u_i \in \mathbb{R}$  and  $p \in \mathcal{M}$  one obtains the linear system

$$a(u_h, \varphi_j) = a \left( \sum_{i \in \mathcal{I}_v} u_i \cdot \varphi_i, \varphi_j \right) = \sum_{i \in \mathcal{I}_v} u_i \cdot a(\varphi_i, \varphi_j) = \ell_f(\varphi_j)$$

which can be written as

$$\mathbf{S}\mathbf{U} = \mathbf{B}, \quad (3.7)$$

where  $\mathbf{U} = (u_i)_{i \in \mathcal{I}_v} \in \mathbb{R}^{|\mathcal{I}_v|}$  denotes the coefficient vector,  $\mathbf{S}_{ij} = a(\varphi_i, \varphi_j) \in \mathbb{R}^{|\mathcal{I}_v| \times |\mathcal{I}_v|}$  the stiffness matrix, and  $\mathbf{B}_j = \int_{\mathcal{M}} f \varphi_j \, da \in \mathbb{R}^{|\mathcal{I}_v|}$  the right-hand side.

For our model problems we have to switch to the discrete ansatz space

$$V_h^0 = V_h \cap \left\{ \int_{\mathcal{M}} v_h \, da = 0 \right\}$$

instead of  $V_h$  on the subdivision surface  $\mathcal{M}$ . Again, the reason is that all constant functions lie in the kernel of all problems (3.1) – (3.3). Then, the stiffness matrix for the Laplace-Beltrami problem (3.1) is given by

$$\mathbf{S}_{ij}^\Delta = \int_{\mathcal{M}} \nabla_{\mathcal{M}} \varphi_i \cdot \nabla_{\mathcal{M}} \varphi_j \, da = \sum_{k \in \mathcal{I}_c} \int_{\Delta} \left( \nabla_{\xi} \Phi_i \cdot G^{-1} \nabla_{\xi} \Phi_j \sqrt{\det G} \right) (\xi, k) \, d\xi \quad (3.8)$$

and for the surface bi-Laplacian problem (3.2) we obtain

$$\begin{aligned} \mathbf{S}_{ij}^{\Delta^2} &= \int_{\mathcal{M}} \Delta_{\mathcal{M}} \varphi_i \cdot \Delta_{\mathcal{M}} \varphi_j \, da \\ &= \sum_{k \in \mathcal{I}_c} \int_{\Delta} \left( \frac{\operatorname{div}_{\xi} \left( \sqrt{\det G} G^{-1} \nabla_{\xi} \Phi_i \right) \operatorname{div}_{\xi} \left( \sqrt{\det G} G^{-1} \nabla_{\xi} \Phi_j \right)}{\sqrt{\det G}} \right) (\xi, k) \, d\xi. \end{aligned} \quad (3.9)$$

The right-hand side is given by

$$\mathbf{B}_j = \int_{\mathcal{M}} f \cdot \varphi_j \, da = \sum_{k \in \mathcal{I}_c} \int_{\Delta} ((f \circ X) \cdot \Phi_j) (\xi, k) \, d\xi$$

for a given function  $f : \mathcal{M} \rightarrow \mathbb{R}$ . Let us remark that this differs from the finite element setting in the following way. The function  $f$  is directly defined on the fixed surface  $\mathcal{M}$  and hence does not have to be interpolated as for polygonal meshes that interpolate  $\mathcal{M}$ . The variational formulation of the eigenvalue problem (3.3) consists in finding a discrete solution  $(u_h, \lambda_h) \in V_h^0 \times \mathbb{R}$  such that

$$a(u_h, v_h) = \lambda_h \cdot m(u_h, v_h), \quad \forall v_h \in V_h^0,$$

where  $m(u_h, v_h) = \int_{\mathcal{M}} u_h v_h \, da$  denotes the  $L^2$ -product on  $\mathcal{M}$ . Furthermore, we denote by  $\mathbf{M}$  the mass matrix with

$$\mathbf{M}_{ij} = m(\varphi_i, \varphi_j) = \int_{\mathcal{M}} \varphi_i \cdot \varphi_j \, da = \sum_{k \in \mathcal{I}_c} \int_{\Delta} \Phi_i \Phi_j \sqrt{\det G} \, d\xi. \quad (3.10)$$

We obtain the discrete eigenvalue problem  $\mathbf{S}\mathbf{U} = \lambda_h \mathbf{M}\mathbf{U}$  which can be solved by *inverse vector iteration with projection* [125].

### 3.3. Numerical Integration

During the assembly of the matrix entries (3.8), (3.9) and (3.10) we have to integrate functions on the unit triangle  $\Delta$ . In contrast to the finite element method the area element is not constant on the triangle  $\Delta$  because the embedded triangles are curved. Hence, we cannot integrate the product of the basis functions resp. their derivatives in a preprocessing step and have to perform the integration on the fly for given data. Furthermore, the isogeometric subdivision approach is based on higher order spline

### 3. Isogeometric Subdivision Method

discretizations with basis functions  $\Phi_i$  that are quartic polynomials and thus for the mass matrix the integrand is a polynomial of degree 8 and for the stiffness matrix (3.8) of degree 6 and for (3.9) of degree 4 if we assume that the metric  $G$  is constant on a regular patch. In the presence of extraordinary vertices we cannot even integrate the matrix entries (3.8), (3.9) and (3.10) exactly because of the singular configuration explained in Chapter 2. This implies that we will solve a system

$$\tilde{\mathbf{S}}\mathbf{U} = \tilde{\mathbf{B}}$$

instead of solving (3.7), where  $\tilde{\mathbf{S}}$  approximates  $\mathbf{S}$  and  $\tilde{\mathbf{B}}$  approximates  $\mathbf{B}$ . The associated, modified variational problem reads as follows: Find  $\tilde{u}_h \in V_h^0$  such that

$$\tilde{a}(\tilde{u}_h, v_h) = \tilde{\ell}_f(v_h), \quad \forall v_h \in V_h^0, \quad (3.11)$$

where  $\tilde{a}(\varphi_i, \varphi_j) = \tilde{\mathbf{S}}_{ij}$  and  $\tilde{\ell}_f(\varphi_j) = \tilde{\mathbf{B}}_j$ . If  $\tilde{a}(\cdot, \cdot)$  is uniformly  $V_h$ -elliptic ( $\tilde{\mathbf{S}}$  positive definite) existence and uniqueness of the discrete solution  $\tilde{u}_h$  is ensured (cf. [22], for instance).

**Remark** *Strang's Lemma* (see [132, 22]) provides the following error estimate for the numerical solution  $\tilde{u}_h$  of (3.11)

$$\begin{aligned} \|u - \tilde{u}_h\| &\leq \underbrace{\|u - u_h\|}_{\text{Approximation error (Cea's Lemma)}} + \underbrace{\|u_h - \tilde{u}_h\|}_{\text{Consistency error (Strang's Lemma)}} \\ &\leq \inf_{v_h \in V_h^0} \left( \|u - v_h\| + \sup_{w_h \in V_h^0} \frac{|a(v_h, w_h) - \tilde{a}(v_h, w_h)|}{\|w_h\|} \right) + \sup_{w_h \in V_h^0} \frac{|\ell(w_h) - \tilde{\ell}(w_h)|}{\|w_h\|} \end{aligned}$$

where  $u$  is the continuous solution of (3.5) and  $u_h$  is the discrete solution of (3.6). The last two terms measure the consistency of  $\tilde{a}$  and  $\tilde{\ell}_f$ . Here,  $\|\cdot\| = \|\cdot\|_{H^1}$  for the Laplace-Beltrami problem (3.1) and  $\|\cdot\| = \|\cdot\|_{H^2}$  for the surface bi-Laplacian problem (3.2). In fact, if a scheme with exact integration fulfills  $\inf_{v_h \in V_h} \|u - v_h\| \leq Ch^p$  (with  $p \geq 1$  and a constant  $C$ ), we ask for a numerical quadrature that preserves this order, i.e.

$$\sup_{w_h \in V_h} \left( \frac{|a(u_h, w_h) - \tilde{a}(u_h, w_h)|}{\|w_h\|} + \frac{|\ell(w_h) - \tilde{\ell}(w_h)|}{\|w_h\|} \right) \leq Ch^p.$$

The optimal  $w_h = \sum_{i \in \mathcal{I}_v} w_i \varphi_i \in V_h^0$  for fixed  $h$  is given by  $w_h = \frac{z_h}{\sqrt{a(z_h, z_h)}}$ , where  $z_h \in V_h^0$  is the solution of

$$a(z_h, \varphi_j) = a(u_h, \varphi_j) - \tilde{a}(u_h, \varphi_j), \quad \forall j \in \mathcal{I}_v,$$

with  $\int_{\mathcal{M}} z_h \, da = 0$  and  $z_h = \sum_{i \in \mathcal{I}_v} z_i \varphi_i$ . Figure 3.2 plots the resulting consistency error depending on the mesh size  $h$  of the control mesh. We observe an improved consistency by two orders for the mid-edge rule compared to the barycenter rule for the surface bi-Laplacian problem (3.2).

Hence, in the presence of extraordinary vertices, we will always solve (3.11) instead of (3.7) we just have to approximate it in a sufficient way. Therefore, we choose numerical quadrature for the integration, i.e. we perform a weighted evaluation-based assembly of the previously defined discrete variational formulations (3.8), (3.9) and (3.10). As quadrature schemes we use standard Gaussian, barycentric and mid-edge assembly and discuss the special treatment at extraordinary vertices introduced in [104]. For the mid-edge scheme we pick up an idea from [70] and provide a lookup table

of the basis function values and their derivatives based on the valencies of the adjacent vertices. As advantages we can integrate over edges instead of facets and we do not have to implement the natural parameterization discussed in Chapter 2. Additionally, these lookup tables allow for the simulation with arbitrary input meshes (i.e. more than one extraordinary vertex in a patch) which is essential for real-world applications as considered in Chapter 5.

For a general function  $g : \Omega \rightarrow \mathbb{R}$  the evaluation-based quadrature on  $\Delta$  reads

$$\int_{\Delta} g(\xi) d\xi \approx \sum_{q=1}^K w_q g(\xi^q),$$

where  $\xi_q \in \Delta$  are the quadrature points,  $w_q$  the weights and  $K$  denotes the number of quadrature points. Now, the stiffness matrices, the mass matrix, and the right-hand side are now replaced by the following quadrature-based counterparts

$$\tilde{\mathbf{S}}_{ij}^{\Delta} = \sum_{k \in \mathcal{I}_c} \sum_{q=1}^K w_q \left( \nabla_{\xi} \Phi_i^T G^{-1} \nabla_{\xi} \Phi_j \sqrt{\det G} \right) (\xi^q, k) \quad (3.12)$$

and

$$\tilde{\mathbf{S}}_{ij}^{\Delta^2} = \sum_{k \in \mathcal{I}_c} \sum_{q=1}^K w_q \left( \operatorname{div}_{\xi} \left( \sqrt{\det G} G^{-1} \nabla_{\xi} \Phi_i \right) \operatorname{div}_{\xi} \left( \sqrt{\det G} G^{-1} \nabla_{\xi} \Phi_j \right) \frac{1}{\sqrt{\det G}} \right) (\xi^q, k) \quad (3.13)$$

as well as

$$\tilde{\mathbf{M}}_{ij} = \sum_{k \in \mathcal{I}_c} \sum_{q=1}^K w_q \left( \Phi_i \cdot \Phi_j \sqrt{\det G} \right) (\xi^q, k) \quad (3.14)$$

and

$$\tilde{\mathbf{B}}_j = \sum_{k \in \mathcal{I}_c} \sum_{q=1}^K w_q \left( (f \circ X) \cdot \Phi_j \sqrt{\det G} \right) (\xi^q, k). \quad (3.15)$$

**Gaussian quadrature** For second-order elliptic problems, standard error estimates for finite element schemes with Gaussian quadrature [30, Chapter 4.1] imply that the expected order of convergence, in case of exact integration, is preserved if the quadrature scheme is exact for polynomials of degree  $p = 6$ . Transferring this to fourth-order problems we request exactness for polynomials of degree  $p = 4$ . This suggests to choose a Gaussian quadrature rule  $\text{GA}(p)$  which guarantees this required exactness. For the Laplace-Beltrami equation (3.1)  $K = 12$  and for the surface bi-Laplacian equation (3.2)  $K = 6$  quadrature points have to be taken into account on the reference triangle  $\Delta$  (for symmetric Gaussian quadrature points on triangles see [47] or Appendix A.2).

**Adaptive Gaussian quadrature** Special care is required close to EVs [111], because the basis functions  $\Phi_i$  are no longer polynomials and the second-order derivatives are singular at the EV. Furthermore, the natural parametrization [131] produces only  $C^0$ -surfaces at EVs instead of  $C^1$ -parametrizations.

Because of the structure of the subdivision scheme, the subdivision surface and correspondingly the basis functions  $\Phi_i$  are spline functions on local triangular mesh rings around each EV, cf. Figure 2.4. In [104] this subdivision ring structure was used as an adaptive refinement strategy of the reference triangle  $\Delta$  around the EV and an application of Gaussian quadrature on the resulting adaptive reference mesh to overcome the limitation of the standard Gaussian quadrature on the reference triangle. More explicitly, for triangles with an EV we decompose the associated reference triangle  $\Delta$  into finer triangles  $\Delta_i^l$  ( $l = 1, \dots, L$ ) of level  $L$  (see Figure 3.1, left), perform the corresponding Gaussian quadrature  $\text{GA}(p)$  on all finer triangles  $\Delta_i^l$  and call this adaptive Gaussian quadrature  $\text{AG}(p, L)$ . Hence, the number of quadrature points  $K$  on these adaptively refined triangles depends on  $L$ :  $K = (3 \cdot L + 1) \cdot 12$  for the Laplace-Beltrami equation (3.1) and  $K = (3 \cdot L + 1) \cdot 6$  for the surface bi-Laplacian equation (3.2). It is worth mentioning that this expensive scheme has to be applied only for a small number of triangles around the finitely many EVs. Finally, let us remark that standard Romberg extrapolation does not lead to any improvement of the adaptive Gaussian quadrature due to the lack of smoothness of the integrands in the mesh size parameter and an adaption of the Romberg method for singular problems failed because the type of singularity is not explicitly known. Furthermore, even with a high number of adaptive steps we never integrate exactly because of the infinite sequence of spline rings discussed above.

**Barycenter quadrature** Implementation of the adaptive Gaussian quadrature is a tedious issue. In particular for large scale engineering problems, where the achievable maximal order of consistency is not needed, already the computing cost of the usual Gaussian quadrature is significant. Therefore, reduced quadrature assembly is a common practice when implementing modeling or simulation tools based on NURBS as well as on subdivision surfaces (e.g. [83, 33]). The simplest and for subdivision surfaces widespread quadrature is the barycentric quadrature (BC) with the center point  $\xi^1 = (\frac{1}{3}, \frac{1}{3})$  of the triangle  $\Delta$  and the weight  $w_1 = \frac{1}{2}$ . This rule is applied to regular as well as to irregular triangles and integrates only affine functions exactly. The method is also used in [33] and leads to a reasonable consistency at least in the energy norm, if only a few EVs in the mesh are present (cf. Figure 3.6 for an example with many EVs).

**Mid-edge quadrature** An alternative, which shows superior performance with respect to the achievable convergence rates (cf. Section 3.4), is the mid-edge quadrature (ME) with  $K = 3$  quadrature points at the midpoints of the edges, i.e.  $\xi^1 = (\frac{1}{2}, 0)$ ,  $\xi^2 = (\frac{1}{2}, \frac{1}{2})$  and  $\xi^3 = (0, \frac{1}{2})$  with weights  $w_1 = w_2 = w_3 = \frac{1}{6}$ . The ME quadrature integrates exactly polynomials of degree  $p = 2$ . Additional to the standard facet-based implementation we follow [70] for a direct implementation of (3.12), (3.13), (3.14) and (3.15) with geometry-independent lookup tables of the basis function values and their derivatives on the unit triangle  $\Delta$  at midpoints of the edges. This substantially simplifies the implementation.

Figure 3.1 (right) depicts a sketch of a generic local control mesh with two, possibly extraordinary, vertices. The valence of the first extraordinary vertex  $p_1^0$  is denoted by  $N_1$ , while the valence of the other vertex  $p_2^0$  is denoted by  $N_2$ . These two control points are surrounded by a fan of additional control points  $p_i^0$  ( $i = 3, \dots, N_1 + N_2 - 2$ ) and  $\mathbf{p}^0 = (p_1^0, p_2^0, \dots, p_{N_1+N_2-1}^0, p_{N_1+N_2-2}^0)$  denotes the vectors of all control points. To compute the limit position of the surface  $p^\infty$  and their Jacobian  $(p_{,1}^\infty, p_{,2}^\infty)$  and Hessian  $(p_{,11}^\infty, p_{,12}^\infty, p_{,22}^\infty)$  in local coordinates  $(\xi_1, \xi_2) \in \Delta$  at edge mid-point (edge between  $p_1^0$  and  $p_2^0$ ) we improve the approach in [70]. Subdividing the vector  $\mathbf{p}^0$  corresponds to a matrix-vector multiplication  $\mathbf{p}^1 = S\mathbf{p}^0$  where  $\mathbf{p}^1$  denotes the vector of new control points  $p_0^1, \dots, p_6^1$  and  $S$  the

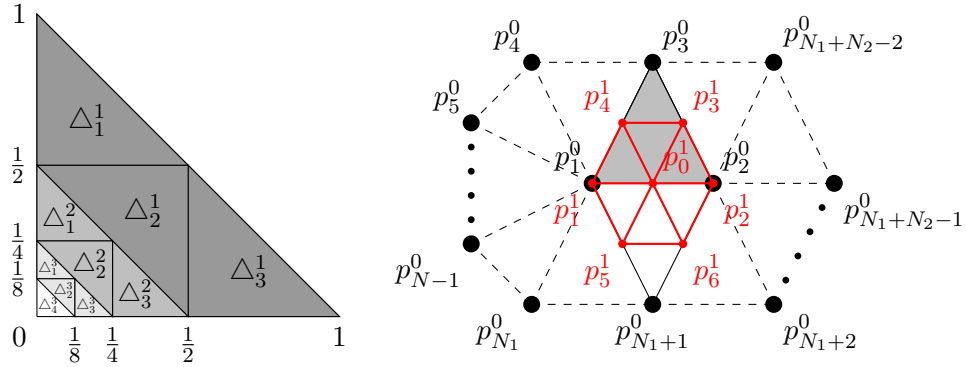


Figure 3.1.: Left: decomposition of the unit triangle  $\triangle$  in smaller triangles corresponding to the spline rings of the adaptive Gaussian quadrature AG( $p, L$ ) for  $L = 3$ . Right: illustration of the vertex configuration two possibly extraordinary vertices  $p_1^0$  of valence  $N_1$  and  $p_2^0$  of valence  $N_2$  surrounded by the one-ring of both vertices. The red lines indicate the new vertices and edges in the 1-neighborhood of  $p_0^1$  after one level of subdivision.

corresponding local subdivision matrix.

In [70] the Taylor expansion (hence the derivatives) is approximated by a polynomial that interpolates limit points on the surface. This approach, which is equivalent to numerical differentiation, provides approximations of the derivatives. Instead, we make use of the well-known exact limit masks for regular vertices [89] and compute the exact limit position  $p^\infty$  and derivatives  $p_{,1}^\infty$ ,  $p_{,2}^\infty$ ,  $p_{,11}^\infty$ ,  $p_{,12}^\infty$  and  $p_{,22}^\infty$  depending on the vertices  $\mathbf{p}^0$  of the coarse mesh. Let us remark that these values can also be easily obtained by evaluating the basis functions and their derivatives in Appendix A.1 at  $\xi = (1/2, 0)$ .

Assume the matrix  $L \in \mathbb{R}^{6 \times 7}$  collects up all limit masks, then

$$\begin{pmatrix} p^\infty \\ p_{,1}^\infty \\ p_{,2}^\infty \\ p_{,11}^\infty \\ p_{,12}^\infty \\ p_{,22}^\infty \end{pmatrix} = L\mathbf{p}^1 = \begin{pmatrix} \frac{1}{2} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} \\ 0 & \frac{1}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -8 & 4 & 0 & 0 & 4 & 0 & 0 \\ -4 & 2 & 2 & -2 & 2 & 2 & -2 \\ -8 & 0 & 4 & 0 & 0 & 4 & 0 \end{pmatrix} \mathbf{p}^1 = LS\mathbf{p}^0,$$

where

$$S = \begin{pmatrix} \frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 \\ \beta_2 & 1 - \beta_2 \cdot N_2 & \beta_2 & 0 & 0 & \dots & 0 & 0 & \beta_2 & \beta_2 & \beta_2 & \dots & \beta_2 & \beta_2 \\ \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{3}{8} & \frac{1}{8} & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 1 - \beta_1 \cdot N_1 & \beta_1 & \beta_1 & \beta_1 & \beta_1 & \dots & \beta_1 & \beta_1 & \beta_1 & 0 & 0 & \dots & 0 & 0 \\ \frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} & \frac{3}{8} & 0 & 0 & \dots & 0 & 0 \\ \frac{1}{8} & \frac{3}{8} & 0 & 0 & 0 & \dots & 0 & 0 & \frac{1}{8} & \frac{3}{8} & 0 & \dots & 0 & 0 \end{pmatrix}$$

for  $N_1, N_2 > 4$ ,  $\beta_1 = \beta(N_1)$ ,  $\beta_2 = \beta(N_2)$  and  $S \in \mathbb{R}^{7 \times N_1 + N_2 - 2}$ . More explicitly, for the hat functions  $\Lambda_i$  (see Chapter 2) associated with a control vertex  $p_i^0$ , the value of the basis function  $\Phi_i(\xi^e, k)$

### 3. Isogeometric Subdivision Method

at the midpoint of the edge  $e \in \mathcal{I}_e$  (connecting  $p_1^0$  and  $p_2^0$  with local coordinates  $\xi^e$  in the triangle  $(\Delta, k)$  corresponding to the vertices  $p_1^0$ ,  $p_2^0$ , and  $p_3^0$ ) is a constant solely depending on  $N_1$  and  $N_2$ . The same holds for the derivatives  $\Phi_{i,1}$ ,  $\Phi_{i,2}$ ,  $\Phi_{i,11}$ ,  $\Phi_{i,12}$  and  $\Phi_{i,22}$  at  $\xi^e$  in the directions on the reference triangle  $\Delta$ . This means that the resulting matrix  $L \cdot S$  is a lookup table for the evaluation of the basis functions and their derivatives for any valencies  $N_1, N_2 > 4$ . For all other cases, i.e.  $N_1 < 5$  or  $N_2 < 5$ , we list the corresponding matrices in appendix A.2. Let us remark that, if  $p_1^0$  and  $p_2^0$  are EVs it is not clear if the global set of basis functions are linearly independent. Here, the corresponding natural parameterization has to be considered in the same way as in [112] to proof this assumption. In all our test cases the resulting isogeometric matrices were linearly independent.

Based on this lookup table approach the *isogeometric subdivision approach* can be implemented by iterating over all edges retrieving values from these lookup tables without implementing the box spline basis functions, their derivatives and the complex subdivision process itself. The iteration over edges instead of elements avoids to evaluate basis function values twice. Furthermore, the involved local matrices in the assembly of the mass and stiffness matrices are smaller for the mid-edge rule than for the barycenter rule. More explicitly, in the regular case, the local IgA–matrices have 144 entries for the BC rule versus 100 for the ME rule. Thus, based on the lookup tables, this leads to an overall faster assembly of mass and stiffness matrices than for the BC rule, even though the number of edges is  $\frac{3}{2}$  times the number of triangles on closed surfaces (e.g. see Table 3.1). The mid-edge assembly process based on lookup tables can easily be generalized to other subdivision schemes like the Catmull–Clark [26] or the Doo–Sabin scheme [46]. Finally, let us remark that the mid-edge assembly can also be performed for many existing finite element codes with a little extension. Here, one needs an additional array of two integer values (corresponding to the element indices) and a boolean, that tells us if this edge (as a pair of the element indices, cf. the definition of the edge index  $\mathcal{I}_e = \mathcal{I}_c \times \mathcal{I}_c$ ) has been already assembled, i.e.  $(i, j, b)$  where  $i, j \in \mathcal{I}_c$  and  $b = 0$  or  $b = 1$ . Then, while iterating over elements, by iterating over the neighbor element one can easily check if the corresponding local matrix has been added to the global matrix, if not, assemble and set  $(i, j, 1)$  and  $(j, i, 1)$ .

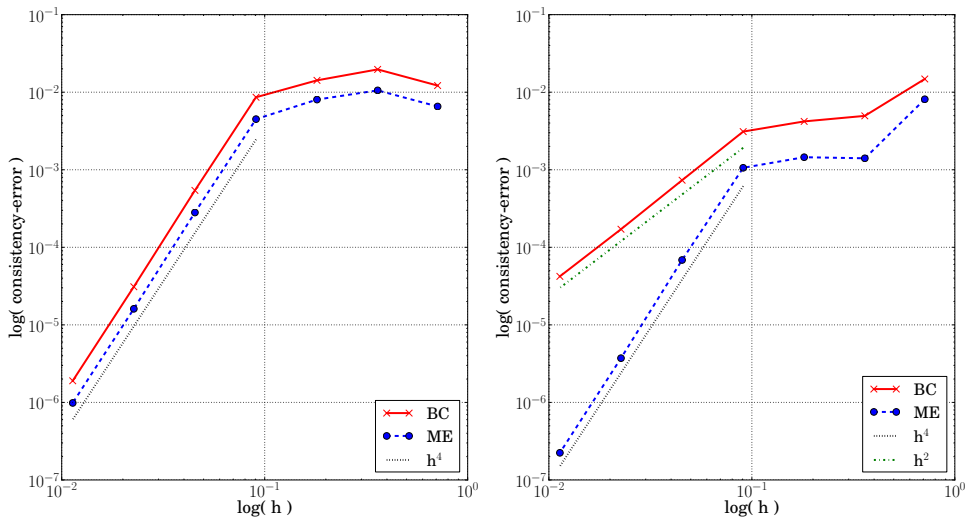


Figure 3.2.: The consistency error is shown in a log-log plot for the Laplacian problem on the torus (cf. Figure 3.3) for varying grid size of the control mesh and for the barycenter quadrature (BC) and the midedge quadrature (ME).



## 3.4. Numerical Experiments

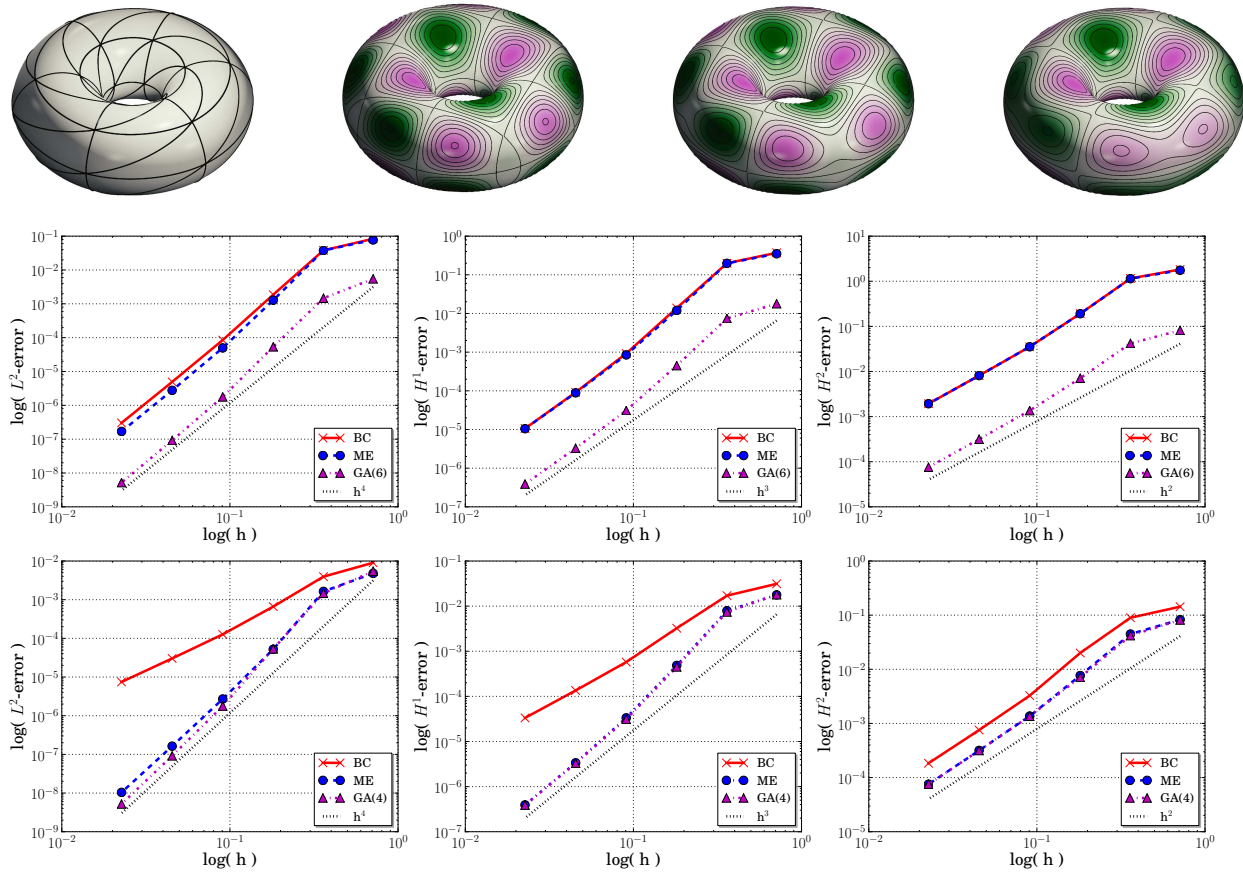





Figure 3.3.: Results for the torus: in the first row from left to right we depict the subdivision limit surface with control lines, isolines and color coding of the right-hand side  $f(y) = \sin(\pi y_1) \sin(\pi y_2) \sin(\pi y_3)$  ( $-1.0$    $1.0$ ), the (numerical reference) solution of the Laplace-Beltrami problem ( $-0.054$    $0.054$ ), and the (numerical reference) solution of the surface bi-Laplacian problem ( $-0.003$    $0.003$ ) on  $\mathcal{M}$ ; in the second and third row log-log plots of the error are reported for the Laplace-Beltrami problem ( $2^{nd}$  row) and the surface bi-Laplacian problem ( $3^{rd}$  row), respectively (from left to right:  $L^2$ -norm,  $H^1$ -semi-norm and  $H^2$ -semi-norm).

We have implemented the proposed methods in C++ and performed tests for four different subdivision surfaces: a torus surface with regular control mesh, a spherical surface (Spherical-3-4) with a control mesh with EVs of valence 3 and 4, a spherical surface (Spherical-5-12) with a control mesh with EVs of valence 5 and 12 and a complex real world hand model where the control mesh has altogether 119 EVs of valence 4, 5, 7, 8, 9 and 10. Let us emphasize that, in the spirit of the isogeometric approach, the control mesh determining the limit subdivision geometry is kept fixed. Furthermore, the limit surface differs from a torus (Figure 3.3) with circular centerline and cross-section or a perfect sphere (Figure 3.5). Then, we successively refine the control mesh using subdivision refinement to improve the accuracy of the discrete PDE solution. To run the simulations for Spherical-3-4, Spherical-

### 3. Isogeometric Subdivision Method

5-12 and the hand model with Stam's parameterization (see Section 2.2) the initial control mesh has to be subdivided at least once to avoid EVs in direct neighborhood. This can be avoided using the mid-edge quadrature with our lookup tables. For comparison, we list only the cases where both approach are possible. In all plots, the mesh size  $h$  refers to the mesh size of these control meshes.

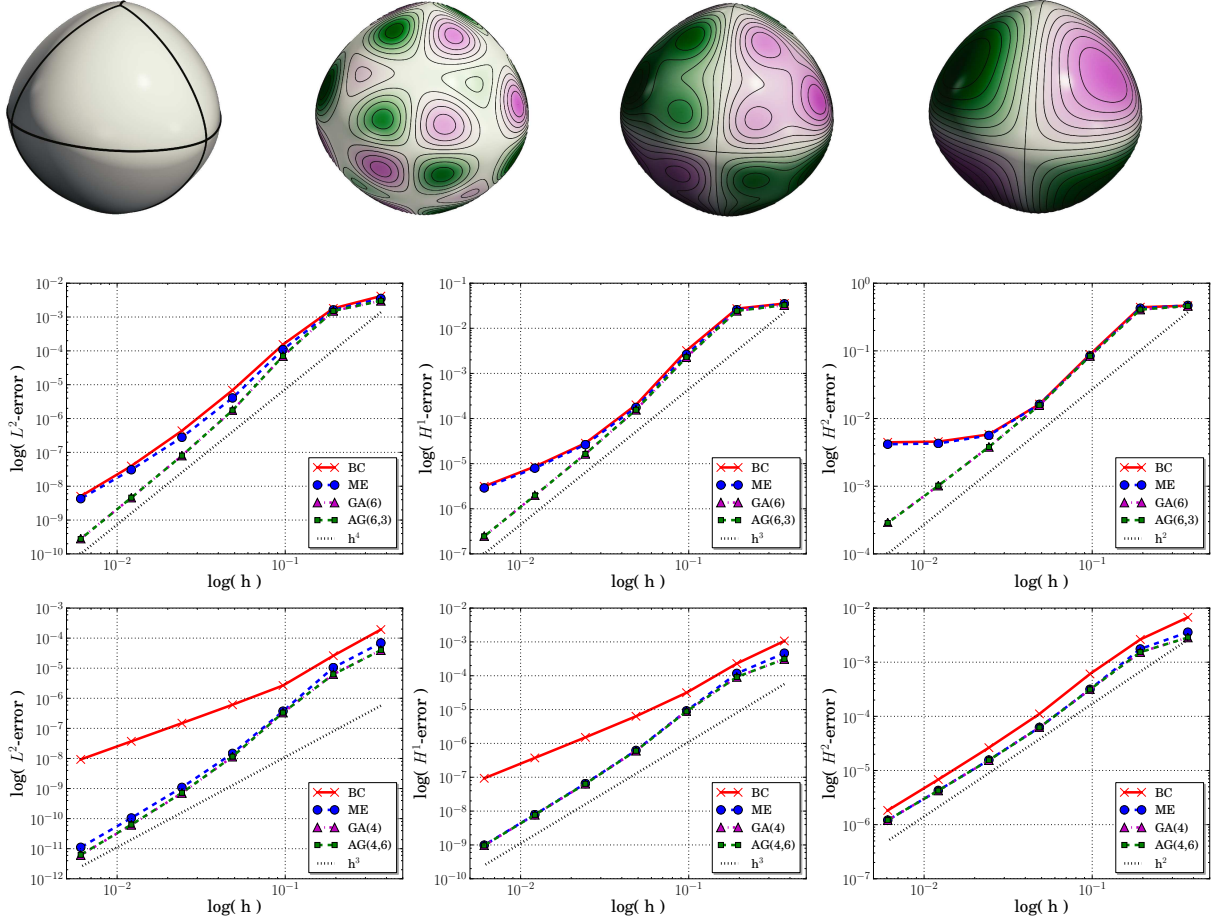


Figure 3.4.: Results for Spherical-3-4: As in Figure 3.3 we show in the top row the limit surface,  $f(y) = \sin(3\pi y_1) \sin(3\pi y_2) \sin(3\pi y_3)$  ( $-1.0$   $1.0$ ), the solution of the Laplace-Beltrami problem ( $-5.25e - 3$   $5.25e - 3$ ), and the solution of the surface bi-Laplacian problem ( $-9.78e - 5$   $9.78e - 5$ ), together with the error-plots for the Laplace-Beltrami problem (2<sup>nd</sup> row) and the surface bi-Laplacian problem (3<sup>rd</sup> row) (from left to right:  $L^2$ -norm,  $H^1$ -semi-norm and  $H^2$ -semi-norm).

On all of these surfaces we solve the two model problems (3.1) and (3.2) for a given right-hand side  $f$  and compare the asymptotic error for different norms: the  $L^2$ -norm, the  $H^1$ -semi-norm and the  $H^2$ -semi-norm. To evaluate the *experimental order of convergence* (eoc), we have computed a reference solution for a control mesh with one additional level of global refinement compared to the finest mesh. Furthermore to compute the reference solution we used the adaptive Gaussian quadrature AG( $p,L$ ) with  $L = 3$  for (3.1) and  $L = 6$  for (3.2) (the estimated convergence rates did not change for larger  $L$ ). As a consequence, the error plots for the finest discretization are less reliable as it becomes apparent

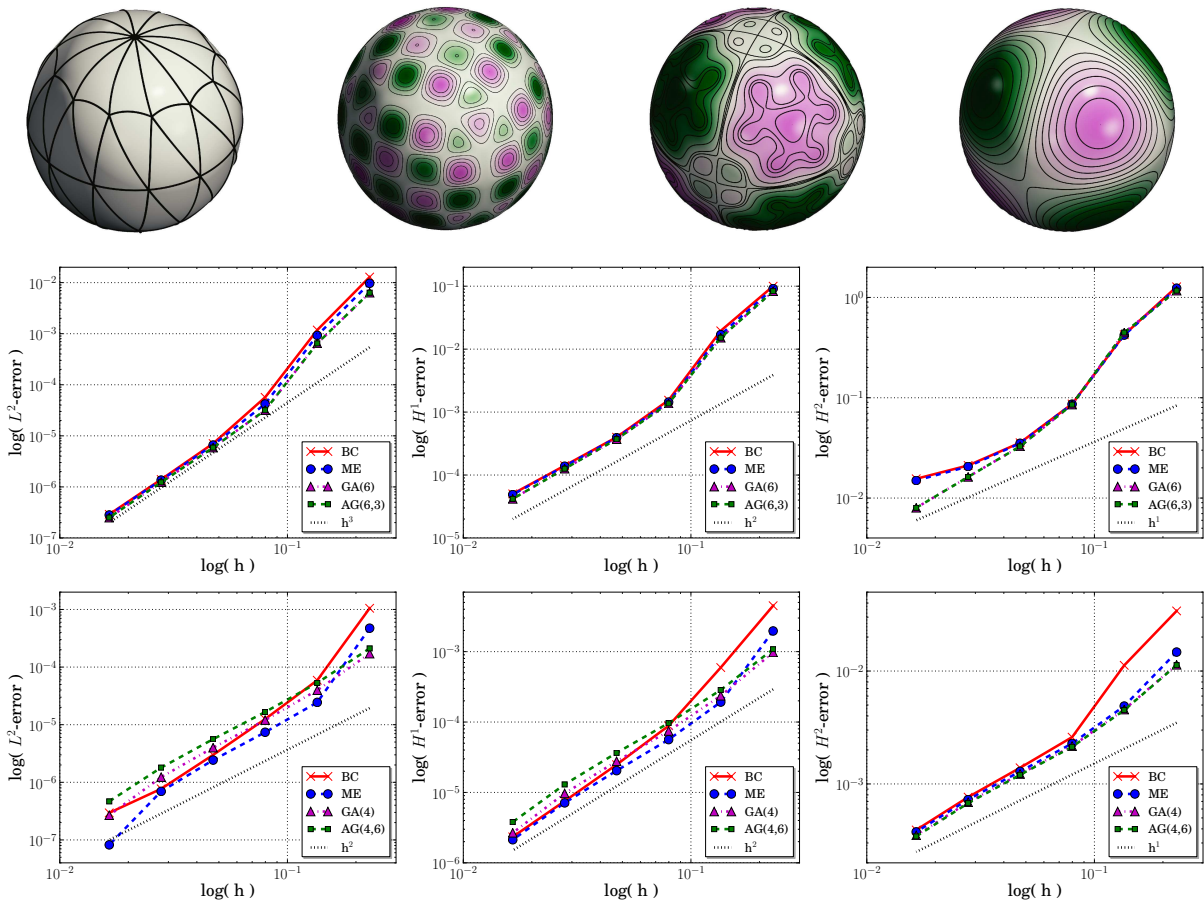


Figure 3.5.: Results for Spherical-5-12: We follow the presentation in Figure 3.4 and plot the limit surface with IgA-element lines, the (same) right-hand side function  $f$  now on this surface, the solution of the Laplace-Beltrami problem ( $-1.58e-2$  to  $1.58e-2$ ), and the solution of the surface bi-Laplacian problem ( $-1.27e-3$  to  $1.27e-3$ ), again together with the error-plots for the Laplace-Beltrami problem (2<sup>nd</sup> row) and the surface bi-Laplacian problem (3<sup>rd</sup> row) (from left to right:  $L^2$ -norm,  $H^1$ -semi-norm and  $H^2$ -semi-norm).

in Figure 3.5 for the  $L^2$ -error plot for problem (3.2) and the spherical shape with low valence EVs (Spherical-3-4).

Figure 3.3 shows our results for the torus with regular control mesh without EVs. All quadrature rules achieve optimal convergence rates (cf. [90]) for the second-order problem (3.1), i.e. 4 in the  $L^2$ -norm, 3 in the  $H^1$ -semi-norm and 2 in the  $H^2$ -semi-norm (with non-adaptive Gaussian quadrature  $GA(p)$ ). Here, “optimal” reflects what we expect for the quartic box spline [90]. For the fourth-order problem (3.2) all quadrature rules achieve optimal rates 4 in the  $L^2$ -norm, 3 in the  $H^1$ -semi-norm and 2 in the  $H^2$ -semi-norm (with non-adaptive Gaussian quadrature  $GA(p)$ ), except the BC rule. In all cases  $GA(p)$  performed best.

The numerical results of the spherical control mesh with EVs of valence 3 and 4 (Spherical-3-4) are depicted in Figure 3.4. For the second-order PDE (3.1) Gaussian quadrature and adaptive Gaussian quadrature achieve the optimal order of convergence. On finer resolutions ME and BC do not achieve

### 3. Isogeometric Subdivision Method

Geometry	$ \mathcal{I}_v $	GA( $p$ )		AG( $p,L$ )		BC		ME	
		$-\Delta_{\mathcal{M}}$	$(-\Delta_{\mathcal{M}})^2$	$-\Delta_{\mathcal{M}}$	$(-\Delta_{\mathcal{M}})^2$	$-\Delta_{\mathcal{M}}$	$(-\Delta_{\mathcal{M}})^2$	$-\Delta_{\mathcal{M}}$	$(-\Delta_{\mathcal{M}})^2$
Spherical-5-12	9218	1.209	1.417	1.374	1.563	0.164	0.272	0.107	0.136
Hand	11586	1.914	2.233	2.615	3.424	0.269	0.460	0.157	0.198

Table 3.1.: Comparison of assembly times for the stiffness matrices (3.12) and (3.13) and the different quadrature rules (Gaussian GA( $p$ ), adaptive Gaussian AG( $p,L$ ), barycenter (BC) and mid-edge (ME)). All times reported in seconds.

optimal convergence rates. GA, AG and ME achieve similar behavior for the fourth-order problem and again the BC method performs worse. The reduced convergence rate in the  $L^2$ -norm (3 instead of 4 in the regular case) for the fourth-order problem is expected to reflect the reduced regularity  $C^1 \cap H^2$  compared to the regular case ( $C^2$ -regularity), which seems to preclude an additional application of an Aubin-Nitsche type argument.

For the control mesh with EVs of valence 5 and 12 (Spherical-5-12) the convergence results are depicted in Figure 3.5 and coincide with our general findings for control meshes with EVs of valence greater than 6. Here, all quadrature schemes show a similar performance, for the Laplace-Beltrami problem (3.1): order 3 in the  $L^2$ -norm, order 2 in the  $H^1$ -semi-norm and order 1 in the  $H^2$ -semi-norm, and for the surface bi-Laplacian problem (3.2): order 2 in  $L^2$ -norm, order 2 in the  $H^1$ -semi-norm and order 1 in the  $H^2$ -semi-norm. This observed loss of approximation order compared to the quartic box spline coincides with the theoretical work by Arden [5] with the reasoning that Loop subdivision functions cannot reproduce cubic polynomials around extraordinary vertices of valence greater than 6.

Finally, we consider the hand model as a complex subdivision surface with many EVs (119 EVs of valence 4, 5, 7, 8, 9 and 10) in Figure 3.6. For both model problems, the Laplace-Beltrami and surface bi-Laplacian problem, the experimental order of converge is not optimal, which suggests that the asymptotic regime of error reduction seems not to be reached even though the reference solution is computed on a mesh with 750k vertices. A reason for this could be the presence of many so-called *polar artifacts* (e.g in Figure 2.3 the subdivision process produces a polar configuration at the north pole). The triangles close to EVs with valence greater than six become much larger during the subdivision process compared to regular regions. If the input configuration (as the valence 10 vertex close to the bottom in Figure 3.6) at the EVs with valence greater than six have already larger triangles than the regular configurations, these errors cannot be compensated during the subdivision process. Here, the one point quadrature does not convergence at all for the surface bi-Laplacian problem which is a serious limitation of the barycenter rule for practical engineering problems (e.g. thin shell problems [33]). In this case the mid-edge quadrature still converges and we considered it as the preferable choice because of robustness and efficiency reasons. As already explained above, the efficiency can even be increased by iterating over edges instead of facets.

Figure 1.1 shows eigenfunctions of the Laplace-Beltrami operator computed via inverse vector iteration with projection. The depicted results underline that the methods discussed so far are also applicable in the numerical eigenmode analysis, which turned out to be an indispensable tool in geometric data analysis and modeling.

Compared to Gaussian quadrature and in particular to the adaptive Gaussian quadrature the mid-edge and the barycenter quadrature are significantly cheaper as reported in Table 3.1. Because of our implementation, the mid-edge rule based on lookup tables and assembly via an edge-iterator performs even better than the (non-optimized) barycenter rule.

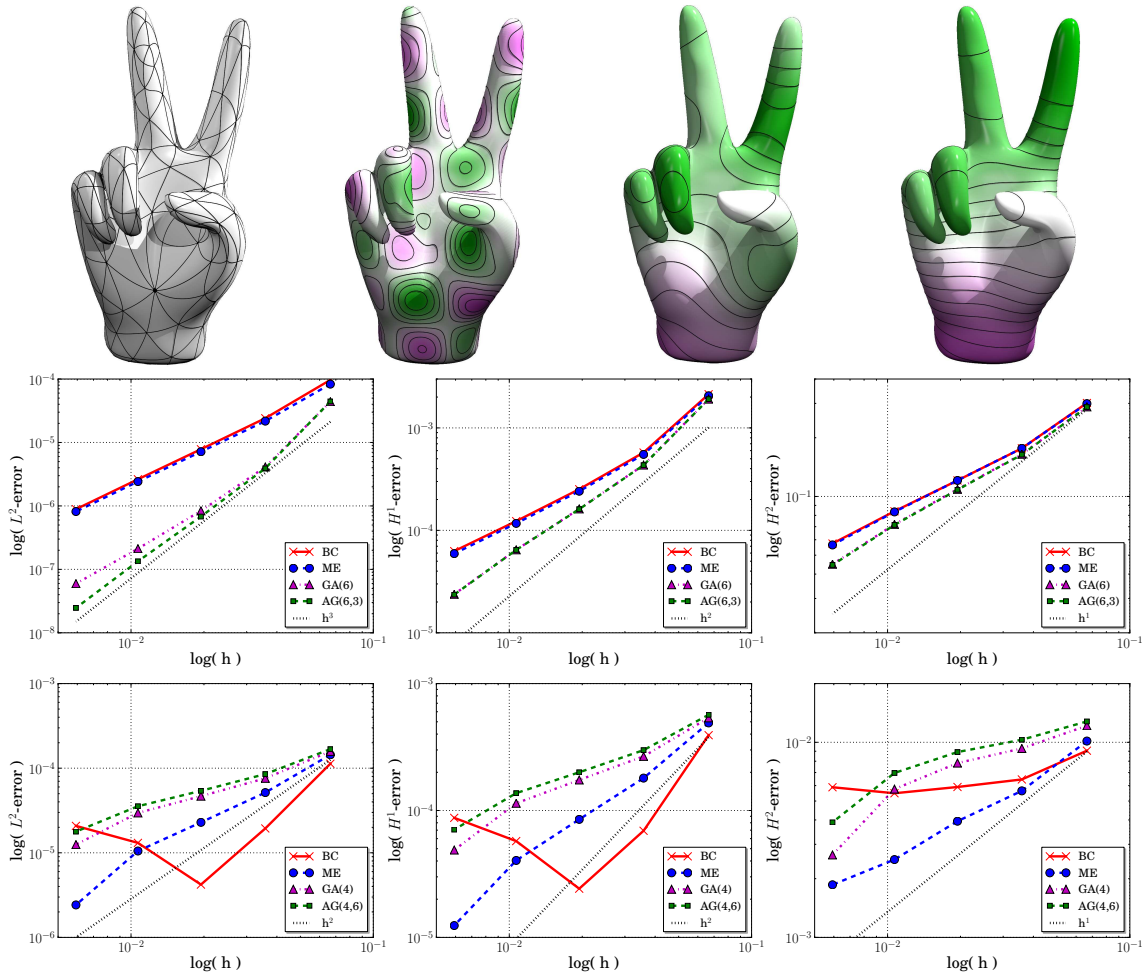


Figure 3.6.: Results for complex hand shape: We follow the presentation in Figure 3.4 and plot the limit surface with IgA-element lines, the (same) right-hand side function  $f$  now on this surface, the solution of the Laplace-Beltrami problem ( $-1.47e - 2$   $\rightarrow$   $1.14e - 2$ ) and the solution of the surface bi-Laplacian problem ( $-8.14e.4$   $\rightarrow$   $7.56e.4$ ), again together with the error-plots for the Laplace-Beltrami problem ( $2^{nd}$  row) and the surface bi-Laplacian problem ( $3^{rd}$  row) (from left to right:  $L^2$ -norm,  $H^1$ -semi-norm and  $H^2$ -semi-norm).



## 4. Parametric Gradient Flows

In this chapter we combine our higher order spatial discretization based on Loop subdivision introduced in Chapter 2 and 3 with *higher order time discretization* based in *implicit Runge–Kutta methods* for the computation of *parametric gradient flows*.

Let  $\{\mathcal{M}(t)\}_{t \geq 0}$  be a family of closed 2-dimensional surfaces  $\mathcal{M} = \mathcal{M}(t)$  embedded in  $\mathbb{R}^3$ , where  $y = y(t)$  denotes a parametrization of  $\mathcal{M}(t) = \mathcal{M}[y]$ . For a given initial surface  $\mathcal{M}_0 = \mathcal{M}(0)$ , metric  $\mathbf{g}_y : \mathcal{T}\mathcal{M} \times \mathcal{T}\mathcal{M} \rightarrow \mathbb{R}$  and energy  $\mathbf{e} : \mathcal{M} \rightarrow \mathbb{R}$  the evolution of  $\mathcal{M}_0$  under

$$y_t = -\text{grad}_{\mathbf{g}_y} \mathbf{e}'[y] \quad (4.1)$$

with  $\mathbf{e}[y(s)] < \mathbf{e}[y(t)] \forall s > t$  denotes the corresponding *gradient flow*, where  $\mathcal{T}\mathcal{M}$  denotes the tangent bundle of  $\mathcal{M}$  and  $y_t$  the partial derivative with respect to time  $t$ .

For the numerical computation we consider the weak formulation of (4.1)

$$\mathbf{g}_y(y_t, \varphi) = -\mathbf{e}'[y](\varphi) \quad \forall \varphi \in \mathcal{V}, \quad (4.2)$$

where  $\mathbf{e}'[y](\varphi)$  denotes the first variation of  $\mathbf{e}[\cdot]$  at  $y$  in direction of a trial function  $\varphi$  and  $\mathcal{V}$  denotes the function space of suitable trial functions. This formulation inherently fulfills the crucial *energy diminishing property* of gradient flows

$$\frac{d}{dt} \mathbf{e}[y(t)] = \mathbf{e}'[y](y_t) = -\mathbf{g}_y(y_t, y_t) < 0, \quad \forall t > 0.$$

These problems typically arise in applications as material sciences, surface processing and modeling, biological membranes, elasticity and physical simulation. Here, during the evolution, the surface often has to fulfill a constraint  $\mathbf{c}[y] = 0$ . Then, the corresponding *constrained gradient flow* reads

$$\mathbf{g}_y(y_t, \varphi) = -\mathbf{e}'[y](\varphi) - \lambda \mathbf{c}'[y](\varphi) \quad \forall \varphi, \quad (4.3a)$$

$$\mathbf{c}'[y](y_t) = 0, \quad (4.3b)$$

where  $\lambda = \lambda(t) \in \mathbb{R}$  denotes the Lagrange multiplier.

Instead of focusing on one particular application, the aim of this chapter is to develop a scheme for a wide range of gradient flows, i.e. for different metrics  $\mathbf{g}_y$  and energies  $\mathbf{e}$ . Hence, we consider a general type of higher order energies

$$\mathbf{e}[y] = \int_{\Omega} r(y, Dy, D^2y), da \quad (4.4)$$

where  $r$  is an arbitrary function of  $y : \Omega \rightarrow \mathbb{R}^3$  with  $\Omega \subset \mathbb{R}^2$ , its Jacobian  $Dy = \left( \frac{\partial y_i(\xi)}{\partial \xi_j} \right)_{ij}$  and its Hessian  $D^2y = \left( \frac{\partial^2 y_i(\xi)}{\partial \xi_k \partial \xi_j} \right)_{ijk}$ . Furthermore, we measure the evolution not only in the  $L^2$ -metric but

consider metrics of higher order derivatives

$$\mathbf{g}_y^{\beta_0, \beta_1}(v, w) = \int_{\mathcal{M}[y]} vw + \beta_0 \nabla_{\mathcal{M}} v : \nabla_{\mathcal{M}} w + \beta_1 \Delta_{\mathcal{M}} v \Delta_{\mathcal{M}} w \, da, \quad (4.5)$$

where  $\beta_0, \beta_1 \in [0, 1]$  and  $\nabla_{\mathcal{M}}$  denotes the surface gradient,  $\Delta_{\mathcal{M}}$  the Laplace-Beltrami operator on  $\mathcal{M}$  and  $A : B = \text{tr}(A^T B)$ . For  $\mathbf{g} = \mathbf{g}_y^{0,0}$  the gradient flow is measured in the  $L^2$ -metric and for  $\mathbf{g} = \mathbf{g}_y^{1,0}$  in the  $H^1$ -metric. Furthermore, we denote by  $H_{\Delta}^2$  the  $\mathbf{g}_y^{1,1}$ -metric. Let us remark that many other metrics fit into the described framework and could be used instead as  $\mathbf{g}$ .

The numerical treatment of parametric gradient flows has been studied extensively in the literature with different approaches for the spatial discretization of (4.2) and (4.3). The most common approach uses standard  $C^0$ -Lagrangian finite elements for energies of type (4.4). For the discretization of the fourth-order PDEs the problem is split into two second-order PDEs, see e.g. [52, 11, 7, 110] for linear and [20, 56] for quadratic finite elements. These splittings are problem dependent and introduce additional degrees of freedom which sometimes need an explicit initial approximation, e.g. approximation of mean curvature [52]. An alternative to overcome the splitting into two problems is to use nonconforming finite elements (e.g. Crouzeix-Raviart elements [138]) or classical  $C^1$ -finite elements. Both approaches offer a direct discretization of (4.2) by losing conformity resp. by introducing additional degrees of freedom with potentially no geometric meaning. In contrast, discrete differential geometry (DDG) seeks to describe continuous geometric objects by discrete counterparts instead of discretizing these geometric objects, see e.g. [18, 38]. In this paper we use *subdivision (spline) surfaces* for the spatial discretization of (4.2) and (4.3). The discretization space is  $H^2$ -conforming, where higher order smoothness is achieved by increasing the support of the basis functions instead of introducing additional degrees of freedom. Subdivision surfaces have already been applied to parametric gradient flow problems, e.g. in [59].

During the evolution of a gradient flow system the energy decreases exponentially in time. Hence, in the time discrete setting large time steps can result in huge deformations of the surface. These problems typically result in stiff situations and explicit time integrators are no longer appropriate. Semi-implicit methods, which solve a linear system for each time step, improve the situation but implicit time integrators allow for much larger time steps (see [108, 107, 7, 110]). Since we discretize in space with higher order elements an implicit Euler time discretization would lead to a time-error dominated scheme. For the fully discrete scheme we expect that the error behaves like

$$\max_{m \in \mathbb{N}} \|y(t_k) - Y_k\|_{L^2} \leq C(\tau^q + h^p)$$

(this is not proven here!), where  $y(t_k)$  denotes the exact solution at time  $t_k$ ,  $Y_k$  its discrete approximation,  $q$  the convergence order of the time scheme and  $p$  the convergence order of the discretization space. For implicit Euler  $q = 1$  the time step size has to scale like  $h^p$  to recover the full higher order spatial approximation power for  $p > 1$ . To overcome this limitation we consider higher order implicit Runge-Kutta methods [75]. Here, the crucial characteristic, the energy diminishing property

$$\mathbf{e}[y(t)] \leq \mathbf{e}[y(s)] \quad \forall s < t \quad (4.6)$$

has to be fulfilled by the fully discrete scheme. In Section 4.4 we show that (4.6) holds for the fully discrete scheme under mild time step size restrictions.

We apply the proposed scheme to a variety of geometric gradient flows among mean curvature and Willmore flow. In our presentation we differentiate between second and fourth-order problems



---

because of two reason. First, the time step size restrictions for second resp. fourth-order problems are  $\tau = \mathcal{O}(h^2)$  resp.  $\tau = \mathcal{O}(h^4)$  in the fully discrete setting with explicit time integrators. Here, the natural question is how these restriction transfer to implicit time integrators. Second, the discretization of fourth-order problems with conforming  $C^1$ -elements brings back the beauty of the classical Ritz-Galerkin method as for second-order problems. In particular, we consider energies of type (4.4)

$$\mathbf{e}^{\alpha_0, \alpha_1}[y] = \int_{\mathcal{M}[y]} \alpha_0 + \frac{\alpha_1}{2} \mathbf{h}^2 \, da, \quad (4.7)$$

where  $da = \sqrt{\det(Dy^T Dy)}$  denotes the area element,  $\mathbf{h}$  is the sum of the principle curvatures and  $\alpha_0, \alpha_1 \in [0, 1]$ . For  $\mathbf{e} = \mathbf{e}^{1,0}$  the gradient flow (4.1) measured in  $L^2$ , i.e.  $\mathbf{g}_y = \mathbf{g}_y^{0,0}$ , becomes the well-known *mean curvature flow*, a classical nonlinear 2nd order parabolic evolution problem

$$y_t = \Delta_{\mathcal{M}} y = -\mathbf{h} n,$$

where  $n = n[y]$  denotes the normal field and  $\Delta_{\mathcal{M}}$  the Laplace-Beltrami operator. For a detailed discussion of the numerical treatment of mean curvature flow see [50, 51, 41]. For  $\mathbf{e} = \mathbf{e}^{0,1}$  the gradient flow (4.1) measured in  $L^2$  is known as the Willmore flow, a classical nonlinear 4th order parabolic evolution problem

$$y_t = \left( \Delta_{\mathcal{M}} \mathbf{h} + \mathbf{h} (|\mathbf{S}|_2^2 - \frac{1}{2} \mathbf{h}^2) \right) n$$

where  $\mathbf{S}$  denotes the shape operator on  $\mathcal{M}$  and  $|\cdot|_2$  the Frobenius norm on the space of endomorphisms on the tangent bundle  $\mathcal{T}\mathcal{M}$ . To apply the scheme in the context of constraint gradient systems we add volume constraints

$$\mathbf{c}[y] = \frac{1}{3} \int_{\mathcal{M}[y]} y \cdot n \, da - V_0 \quad (4.8)$$

where  $V_0 = \frac{1}{3} \int_{\mathcal{M}_0} y_0 \cdot n[y_0] \, da$ .

The finite element approximation of Willmore flow was first investigated by Rusu [124] based on a mixed method for the surface parametrization  $x$  and the mean curvature vector  $\mathbf{h} n$  as independent variables. In [36] the scheme was applied to the problem of surface restoration. Deckelnick and Schieweck established convergence of a  $C^1$ -conforming finite element approximation for axial symmetric surfaces [42], i.e. applying one-dimensional  $C^1$ -elements. An alternative scheme, which in particular ensures a better distribution of nodes on the evolving surface was presented by Barrett, Garcke and Nürnberg [9, 10]. In [20] Bonito, Nochetto and Pauletti develop a novel vector formulation and derive a scheme based on quadratic elements for the simulation of biomembranes. Again, for the simulation of biomembranes Feng and Klug apply subdivision surfaces in [59]. Ohlischläger and Rumpf [108, 107, 7] derive a novel nested time discretization of Willmore flow and apply this to surface restoration and blending. In [110] the scheme was extended to the corresponding anisotropic flow.

This chapter is organized as follows. In Section 4.1 we review the nested time discretization for anisotropic Willmore flow. In Section 4.2 we apply the spatial discretization derived in Chapter 2 to the energies and metrics. The higher order implicit Runge-Kutta time discretization is described in Section 4.3. In Section 4.4 we prove that our scheme preserves the crucial energy diminishing property and discuss the time step restrictions based on our space discretization. Then, the fully discrete algorithm is

described in Section 4.5. Finally, in Section 4.6 various computational results are presented. Appendix A.3 lists energies and metrics and its discrete variations to facilitate the implementation.

## 4.1. Excursion: Nested Variational Time Discretization for Willmore Flow

In this section we will briefly recall the nested time discretization of anisotropic Willmore flow as proposed in [110]. The approach of nested variational time discretization of isotropic Willmore flow was introduced in [108, 107, 7] to overcome the limitations of semi-implicit time discretizations. The time discretization proposed in [7] builds upon the general paradigm of *natural time discretization*. In the context of geometric flows, this approach was studied by Luckhaus and Sturzenhecker [98] leading to a fully implicit variational time discretization for mean curvature motion in the space of functions of bounded variation and by Chambolle [28], who reformulated this scheme in terms of a level set method and generalized it for the approximation of anisotropic mean curvature motion in [15, 27]. The approach of [7] was adapted in [110] to the time discretization of the anisotropic Willmore flow which is fully consistent with Finsler geometry.

**Natural time discretization** For a given initial surface  $\mathcal{M}_0 = \mathcal{M}[y_0]$  with parameterization  $y_0$ , we define a time discrete family  $(y_k)_{k=0, \dots}$  with the desired property  $y_k \approx y(k\tau)$  for the given time step size  $\tau$ . Then, we successively solve a sequence of variational problems

$$y_{k+1} = \arg \min_y \text{dist}(y_k, y)^2 + 2\tau e[y], \quad (4.9)$$

where

$$\text{dist}(y_k, y) = \inf_{c \in \Gamma[y_k, y]} \int_0^1 \sqrt{\mathbf{g}_{c(s)}(\dot{c}(s), \dot{c}(s))} ds$$

denotes the Riemannian distance of  $y$  from  $y_k$  on the manifold (known as the path length in Riemannian geometry) and

$$\Gamma[y_k, y] = \{c \in C^1 \mid c(0) = y_k, c(1) = y\}$$

is the set of smooth curves  $c$  connecting  $y_k$  with  $y$ . This generalized approach to gradient flows was developed to describe classical gradient flows in a different setup, e.g. the classical heat flow is known to be the  $L^2$ -gradient flow of the Dirichlet energy but it is also the  $L^2$ -Wasserstein gradient flow of the entropy functional (cf. [85, 109]). Here, the striking observation is that one immediately obtains the energy diminishing property, i.e.

$$e[y_{k+1}] + \frac{1}{2\tau} \text{dist}(y_k, y_{k+1})^2 \leq e[y_k].$$

Typically, in the context of geodesics in Riemannian geometry, one studies the path energy instead path length

$$\text{sqrdist}(y_k, y) = \inf_{c \in \Gamma[y_k, y]} \int_0^1 \mathbf{g}_{c(s)}(\dot{c}(s), \dot{c}(s)) ds.$$

The reason is that the treatment is easier and the Cauchy-Schwartz inequality gives us

$$\text{dist}(y_k, y)^2 \leq \text{sqrdist}(y_k, y),$$

where equality holds if and only if  $\mathbf{g}_{c(s)}(\dot{c}(s), \dot{c}(s))$  is constant. Then, the natural time discretization (4.9) becomes

$$y_{k+1} = \arg \min_y \text{sqrdist}(y_k, y) + 2\tau e[y].$$

Again, we immediately obtain the energy diminishing property

$$e[y_{k+1}] + \frac{1}{2\tau} \text{sqrdist}(y_k, y_{k+1}) \leq e[y_k].$$

For a detailed discussion on this topic see [107]. This formulation gives us classical discretization schemes, e.g. for  $\mathbf{g} = \mathbf{g}^{0,0}$  and  $e[y] = e^{1,0}[y]$  we obtain the well-known implicit Euler method for mean curvature flow

$$y_{k+1} = \arg \min_y \int_{\mathcal{M}[y_k]} (y_k - y)^2 da + 2\tau \int_{\mathcal{M}[y]} da.$$

**Finsler geometry** Finsler manifolds are the extensions of Riemannian manifolds where the norm is not necessarily induced by an inner product. For a comprehensive introduction to general Finsler geometry see [8] and for the special topic of gradient flows see [103, 115]. Let us assume that  $\gamma : \mathbb{R}^{d+1} \rightarrow [0, \infty)$  is a sufficiently smooth function in co-dimension one, where  $d = 1, 2$ . Furthermore, we assume that  $\gamma$  is

- positive,
- 1-homogeneous (i.e.  $\gamma(\lambda p) = |\lambda| \gamma(p)$  for all  $\lambda \in \mathbb{R}, p \in \mathbb{R}^{d+1}$ ) and
- satisfies the ellipticity condition

$$\gamma''(p)qq \geq c_0 \|q\|^2 \quad \forall p, q \in \mathbb{R}^{d+1}, \|p\| = 1, p \cdot q = 0$$

for some positive constant  $c_0$  and the Euclidean norm  $\|\cdot\|$ .

The function  $\gamma(n)$  represents the anisotropic area weight for a surface normal  $n$  such that the anisotropic area functional becomes

$$\mathbf{a}[y] = \int_{\mathcal{M}[y]} \gamma(n[y]) da = \int_{\Omega} \gamma(n[y]) \sqrt{\det g[y]} d\xi = \int_{\Omega} \gamma(\tilde{n}[y]) d\xi, \quad (4.10)$$

where  $\gamma(n[x]) da$  denotes the anisotropic area element and  $\tilde{n} = y_{,1} \times y_{,2}$  the unnormalized normal field to  $y$ . We define the dual function of  $\gamma$  as

$$\gamma^*(x) = \sup\{\langle x, \psi \rangle \mid \psi \in B_\gamma\} \quad \forall x \in \mathbb{R}^{d+1},$$

where  $B_\gamma$  denotes the unit Ball in the  $\gamma$ -norm. The ellipticity assumption ensures that  $(\mathbb{R}^{d+1}, \gamma)$  and its dual space  $(\mathbb{R}^{d+1}, \gamma^*)$  are uniformly convex Banach spaces and the duality map  $T : (\mathbb{R}^{d+1}, \gamma^*) \rightarrow$

$(\mathbb{R}^{d+1}, \gamma)$ , with

$$T(x) = \frac{1}{2} \partial(\gamma^*(x)^2)$$

is an odd single-valued bijective continuous map. More precisely  $T(0) = 0$ ,

$$T(x) = \gamma^*(x) \nabla \gamma^*(x)$$

for  $x \neq 0$ , and

$$T^{-1}(\xi) = \gamma(\xi) \nabla \gamma(\xi)$$

for  $\xi \neq 0$ . For details we refer to [115]. Now, based on the anisotropy  $\gamma$  and its dual  $\gamma^*$  we define an anisotropic distance  $\text{sqrdist}$  of a manifold  $\mathcal{M}[y]$  from a manifold  $\mathcal{M}[x]$  by

$$\text{sqrdist}(\mathcal{M}[x], \mathcal{M}[y]) = \int_{\mathcal{M}[x]} \gamma^*(y-x)^2 \gamma(n[x]) \, da \quad (4.11)$$

for parameterizations  $x$  and  $y$ . The isotropic case is recovered by choosing  $\gamma(\cdot) = \|\cdot\|$ . The unit ball  $\mathcal{F} = \{x \in \mathbb{R}^{d+1} : \gamma(x) \leq 1\}$  in  $(\mathbb{R}^{d+1}, \gamma)$  is called the *Frank diagram*, the associated dual unit ball  $\mathcal{W} = \{x \in \mathbb{R}^{d+1} : \gamma^*(x) \leq 1\}$  is the corresponding *Wulff shape*. Clarenz [35] showed that Wulff shapes are the minimizers of the corresponding anisotropic Willmore functional. For a detailed discussion of anisotropic functions in the context of gradient flows we refer to [103].

**Nested time discretization** Based on these considerations let us first consider anisotropic mean curvature flow, which is defined as the gradient flow of the anisotropic area functional (4.10) with respect to the anisotropic metric (4.11). In this case the variational time discretization is associated with the minimization of

$$\begin{aligned} e_{\text{in}}[x, y] &= \text{sqrdist}(\mathcal{M}[x], \mathcal{M}[y]) + 2\tilde{\tau} \int_{\mathcal{M}[y]} \gamma(n[y]) \, da \\ &= \int_{\mathcal{M}[x]} \gamma^*(y-x)^2 \gamma(n[x]) \, da + 2\tilde{\tau} \int_{\mathcal{M}[y]} \gamma(n[y]) \, da, \end{aligned}$$

with respect to  $y$  for a given surface  $\mathcal{M}[x]$  and  $\tilde{\tau} > 0$ . Let us denote by  $y[x]$  the minimizer for a given surface parameterization  $x$ . By computing the variation of  $e_{\text{in}}[x, y]$  with respect to  $y$  we obtain

$$\begin{aligned} 0 &= \int_{\mathcal{M}[x]} (\gamma^*(y-x) \nabla \gamma^*(y-x) \cdot \theta) \gamma(n[x]) \, da + \tilde{\tau} \mathbf{a}'[y](\theta) \\ &= \tilde{\tau} \int_{\mathcal{M}[x]} T\left(\frac{y-x}{\tilde{\tau}}\right) \cdot \theta \gamma(n[x]) \, da + \tilde{\tau} \mathbf{a}'[y](\theta) \end{aligned}$$

for smooth test functions  $\theta : \mathcal{M}[x] \rightarrow \mathbb{R}^{d+1}$ . Together with

$$y_t(k\tilde{\tau}) \approx \frac{y-x}{\tilde{\tau}}$$

this reflects the weak formulation of anisotropic mean curvature flow given by

$$\int_{\mathcal{M}[y]} T(y_t) \cdot \theta \gamma(n[y]) da = -\mathbf{a}'[y](\theta) = \int_{\mathcal{M}[y]} \mathbf{h}_\gamma[y] \frac{n[y]}{\gamma(n[y])} \cdot \theta \gamma(n[y]) da, \quad (4.12)$$

for a parametrization  $y$  and smooth test functions  $\theta$  defined on  $\mathcal{M}[y]$ , where

$$\mathbf{h}_\gamma[y] = \operatorname{div}_{\mathcal{M}[y]}(\nabla \gamma(n[y]))$$

denotes the anisotropic mean curvature (see [34] and [115]). Thus, from (4.12) we deduce that

$$T(y_t) = -\mathbf{h}_\gamma[y] \frac{n[y]}{\gamma(n[y])}$$

or equivalently we achieve the strong formulation of anisotropic mean curvature flow

$$\begin{aligned} y_t &= T^{-1} \left( -\mathbf{h}_\gamma[y] \frac{n[y]}{\gamma(n[y])} \right) \\ &= \gamma \left( -\mathbf{h}_\gamma[y] \frac{n[y]}{\gamma(n[y])} \right) \nabla \gamma \left( -\mathbf{h}_\gamma[y] \frac{n[y]}{\gamma(n[y])} \right) \\ &= -\frac{\mathbf{h}_\gamma[y]}{\gamma(n[y])} \gamma(n[y]) \nabla \gamma(n[y]) \\ &= -\mathbf{h}_\gamma[y] \nabla \gamma(n[y]) = \kappa_\gamma[y], \end{aligned} \quad (4.13)$$

where the last equality holds due to the 1-homogeneity of  $\gamma$ . This corresponds to the isotropic case where  $\gamma(\cdot) = \|\cdot\|$  and  $y_t = -\mathbf{h}[y] n[y]$ .

Next, we deal with the actual anisotropic Willmore flow and consider the anisotropic Willmore functional defined as follows for a parametrization  $x$  of  $\mathcal{M}[x]$

$$\mathbf{w}[x] = \frac{1}{2} \int_{\mathcal{M}[x]} \mathbf{h}_\gamma[x]^2 \gamma(n[x]) da.$$

Then the abstract variational time discretization of anisotropic Willmore flow reads as follows:

Given  $\mathcal{M}[x_k]$  and time step  $\tau$  find a mapping  $x$  such that  $x$  minimizes

$$x_{k+1} = \arg \min_x \operatorname{sqrdist}(\mathcal{M}[x_k], \mathcal{M}[x]) + 2\tau \mathbf{w}[x]$$

First, let us remark that this is the implicit Euler method corresponding to (4.2) where  $\mathbf{e}[x] = \mathbf{w}[x]$ ,  $x_t \approx \frac{x_{k+1} - x_k}{\tau}$  and

$$\begin{aligned} \operatorname{sqrdist}(\mathcal{M}[x_k], \mathcal{M}[x]) &= \mathbf{g}_{x_k} \left( \frac{x_{k+1} - x_k}{\tau}, \frac{x_{k+1} - x_k}{\tau} \right) \\ &= \int_{\mathcal{M}[x_k]} \gamma^* \left( \frac{x_{k+1} - x_k}{\tau} \right)^2 \nabla \gamma(n[x_k]) da. \end{aligned}$$

Again, for discretization of this functional we need  $C^1$ -elements. We will now replace the anisotropic mean curvature vector  $\kappa_\gamma$  by the discrete speed extracted from a scheme for a single time step of

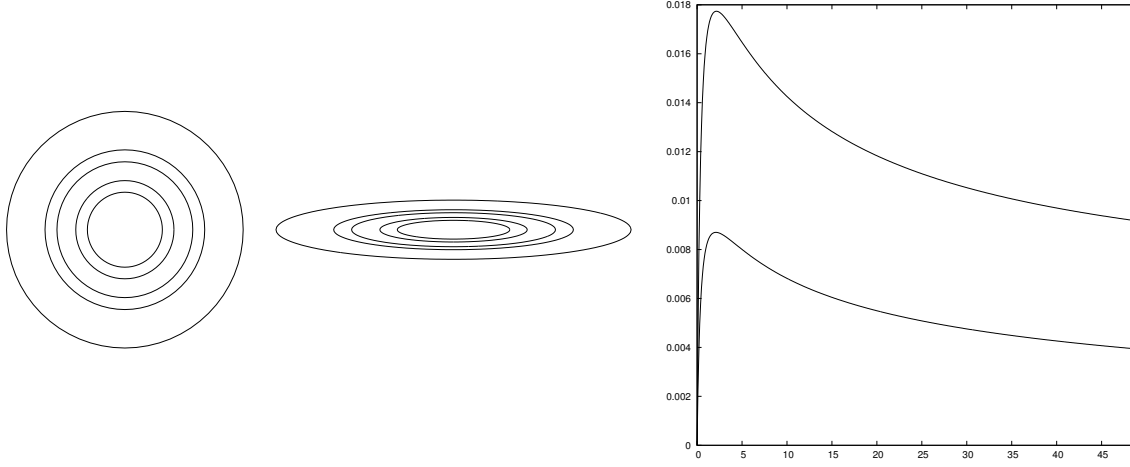


Figure 4.1.: The evolution of a unit circle under isotropic Willmore flow is plotted on the left. For the computation we used as initial grid size  $h = 0.0981$  resulting from 64 vertices. Furthermore,  $\tau = h$ ,  $\tilde{\tau} = h^2$  and the resulting discrete curves are shown for  $t = 0, 10\tau, 50\tau, 100\tau, 500\tau$ . In the middle we display the evolution of an ellipse (with half axes 6 and 1) under anisotropic Willmore flow with 256 elements and  $h = 0.0984$ . Here, we consider  $\tau = h$ ,  $\tilde{\tau} = h^2$  and display the approximate solutions for  $t = 0, 10\tau, 50\tau, 100\tau, 500\tau$ . Next, the associated  $L^2$ -errors are plotted over time on the right, where the lower error curve corresponds to the evolution results on the left.

anisotropic curvature flow (4.12). First, we use that  $\gamma$  is 1-homogeneous and  $\gamma^*(\nabla\gamma(\xi)) = 1$  for all  $\xi \in \mathbb{R}^{d+1}$  such that

$$\gamma^*(\kappa_\gamma)^2 = \gamma^*(-\mathbf{h}_\gamma \nabla\gamma(n))^2 = \mathbf{h}_\gamma^2 \gamma^*(-\nabla\gamma(n))^2 = \mathbf{h}_\gamma^2.$$

Then, we obtain

$$\mathbf{w}[x] = \frac{1}{2} \int_{\mathcal{M}[x]} \mathbf{h}_\gamma[x]^2 \gamma(n[x]) da = \frac{1}{2} \int_{\mathcal{M}[x]} \gamma^*(\kappa_\gamma[x])^2 \gamma(n[x]) da.$$

Second, we apply the approximated speed

$$y_t(k\tilde{\tau}) \approx \frac{y-x}{\tilde{\tau}}$$

to (4.13) such that

$$\frac{y-x}{\tilde{\tau}} \approx \kappa_\gamma[y]$$

and hence

$$\mathbf{w}[x] \approx \frac{1}{2\tilde{\tau}^2} \int_{\mathcal{M}[x]} \gamma^*(y-x)^2 \nabla\gamma(n[x]) da.$$

Finally, based on this approximation we derive the actual time discretization of anisotropic Willmore flow. For a given surface parametrization  $x_k$  of the surface  $\mathcal{M}[x_k]$  at a time step  $k$  we define the outer

$n$	$h(t)$	$L^2$ -error		$h(t)$	$L^2$ -error	
		$(\tau = \tilde{\tau} = h_0^2)$	$eoc$		$(\tau = \tilde{\tau} = h_0)$	$eoc$
4	4.166e-1	4.830e-3		4.482e-1	1.916e-2	
5	2.096e-1	1.328e-3	1.879	2.258e-1	1.087e-2	0.826
6	1.049e-1	3.403e-4	1.969	1.132e-1	5.804e-3	0.909
7	5.249e-2	8.561e-5	1.992	5.668e-2	3.000e-3	0.954
8	2.625e-2	2.144e-5	1.998	2.836e-2	1.525e-3	0.977

Table 4.1.: The  $L^2$ -error between the exact solution of the self-similar evolution of circles under Willmore flow and the discrete solution of the fully implicit variational time discretization is listed at time  $t = 0.1542$  for a grid size  $h(t)$  (left) and  $t = 0.3927$  (right). On the left we consider time step sizes  $\tau$  and  $\tilde{\tau}$  of the order of the squared spatial grid size  $h_0$  at the initial time  $t = 0$ , whereas on the right both time step sizes are taken equal to the grid size  $h_0$ . In both cases we have considered  $2^n$  vertices for the polygon, resulting in an initial grid size  $h_0 = \frac{2\pi}{2^n}$ .

functional

$$e_{\text{out}}[x_k, x, y] = \int_{\mathcal{M}[x_k]} \gamma^*(x - x_k)^2 \gamma(n[x_k]) da + \frac{\tau}{\tilde{\tau}^2} \int_{\mathcal{M}[x]} \gamma^*(y - x)^2 \gamma(n[x]) da,$$

and we end up with the following fully nonlinear variational time discretization of anisotropic Willmore flow:

Given an initial surface  $\mathcal{M}[x_0]$  with parametrization  $x_0$  we define a sequence of surfaces  $\mathcal{M}[x_k]$  with parametrizations  $x_k$  for  $k = 1, \dots$  via the solution of the following sequence of nested variational problems

$$\begin{aligned} x_{k+1} &= \arg \min_x e_{\text{out}}[x_k, x, y[x]], & \text{where} \\ y[x] &= \arg \min_y e_{\text{in}}[x, y]. \end{aligned} \quad (4.14)$$

It is worth to mention that this variational time discretization does not involve derivatives of the anisotropy  $\gamma$  resp. its dual  $\gamma^*$ . Nevertheless, in the context of the actual computation, differentiation is required to run Newton methods for the associated Lagrangian functional. Indeed, for this we will need  $\gamma, \gamma^* \in C^3(\mathbb{R}^{d+1} \setminus \{0\})$ ; moreover, unless  $(\gamma^*)^2 \in C^3(\mathbb{R}^{d+1})$  (which holds for  $\gamma(p) = \sqrt{Ap \cdot p}$  with a symmetric positive definite matrix  $A$ ), a regularization will be required.

**Numerical results** In [110] the minimization problem (4.14) has been discretized with piecewise affine finite elements based on the approach developed in [107, 7]. This corresponds to the surface finite elements introduced by Dziuk in [49]. For the computation we have to solve an optimization for the associated Lagrangian function as described in [110]. For a general discussion of optimization with Lagrangian functions see [106] and in the context of nested time discretization see [107, 7]. Let us remark that the minimization problem (4.14) could easily be discretized with Loop's subdivision surfaces.

Now, we show applications of the proposed algorithm to the evolution of curves in  $\mathbb{R}^2$  under anisotropic

$n$	$h(t)$	$L^2$ -error ( $\tau = \tilde{\tau} = h_0^2$ )	$eoc$	$h(t)$	$L^2$ -error ( $\tau = \tilde{\tau} = h_0$ )	$eoc$
5	1.435e+0	1.648e-1		1.274e+0	1.942e-1	
6	6.487e-1	3.476e-2	1.960	5.875e-1	7.089e-2	1.303
7	3.069e-1	8.762e-3	1.841	2.842e-1	3.424e-2	1.002
8	1.525e-1	2.182e-3	1.987	1.396e-1	1.724e-3	0.966

Table 4.2.: As in Table 4.1 experimental orders of convergence are reported, now for the self-similar evolution of the ellipses (with half axis 6 and 1) under anisotropic Willmore flow. Here, again polygons with  $2^n$  vertices are considered, equi-distributed along the initial ellipse with an initial grid size  $h_0 = \frac{24.172}{2^n}$ . On the left the error is evaluated at time  $t = 0.596576$  and on the right at time  $t = 0.77238$ .

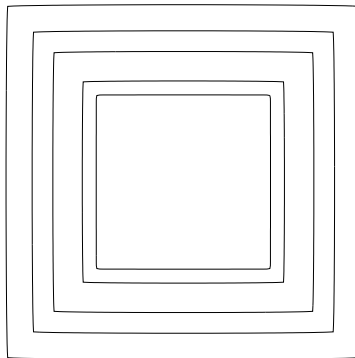


Figure 4.2.: Evolution of the unit sphere with respect to the regularized  $\ell^\infty$ -norm under anisotropic Willmore flow for the anisotropy  $\|\cdot\|_{\ell^\infty_\varepsilon}$  with  $\varepsilon = 0.0001$ . For this computation we consider 200 vertices leading to an initial grid size  $h_0 = 0.04$ . Furthermore,  $\tau = h_0$  and  $\tilde{\tau} = h_0^2$  and the resulting discrete curves are shown for  $t = 0, 10\tau, 50\tau, 100\tau, 200\tau$ .



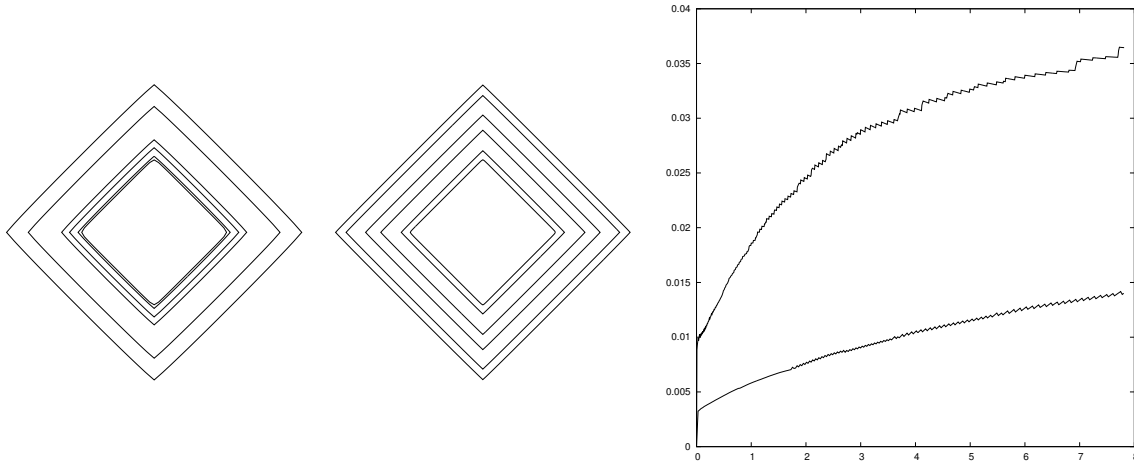


Figure 4.3.: Evolution of the unit sphere with respect to the regularized  $\ell^1$ -norm under anisotropic Willmore flow for the anisotropy  $\gamma(\cdot) = \|\cdot\|_{\ell^\infty}$ . The parameters are  $h_0 = 0.0078$ ,  $\varepsilon = 0.001$ ,  $\tau = \tilde{\tau} = h_0^2$  and curves are plotted at times  $t = 0, 10\tau, 50\tau, 100\tau, 500\tau, 1000\tau$  on the left and  $h_0 = 0.0283$ ,  $\varepsilon = 0.0001$ ,  $\tau = h_0$ ,  $\tilde{\tau} = h_0^2$ ,  $t = 0, 10\tau, 50\tau, 100\tau, 200\tau, 275\tau$  in the middle. On the right the associated  $L^2$ -errors are plotted over time, where the lower error curve corresponds to the evolution results on the left.

Willmore flow, i.e.  $d = 1$ .

Beside anisotropies with ellipsoidal Wulff shapes we study regularized crystalline anisotropies based on a suitable regularization. In contrast to previous definitions of anisotropic Willmore flow [107] this formulation results in self-similar growing of Wulff shapes under the corresponding anisotropic Willmore flow like in the isotropic case. A remarkable fact is that time steps up to the order of the spatial grid size are possible, i.e.  $\tau = \mathcal{O}(h)$ , which is caused by the fact that only second-order problems appear compared to the original fourth-order problem. A detailed discussion of the anisotropic functions and its duals considered here can be found in [103].

At first, we study anisotropies of the type

$$\gamma(z) = \sqrt{a_1^2 z_1^2 + a_2^2 z_2^2}$$

for given  $a_1, a_2 > 0$ . In that case the dual anisotropy is given by

$$\gamma^*(z) = \sqrt{\frac{z_1^2}{a_1^2} + \frac{z_2^2}{a_2^2}}.$$

Figure 4.1 compares the evolution of a circle of radius  $R_0 = 1$  under isotropic Willmore flow for  $a_1 = a_2 = 1$  with the evolution of an ellipse with half-axes  $a_1 = 6$  and  $a_2 = 1$  under the corresponding anisotropic flow. In contrast to previous schemes [7], the initial curve  $\mathcal{M}_0$  expands in a self-similar fashion, i.e.  $\mathcal{M}[x(t)] = R(t)\mathcal{M}_0$  with  $R(t) = \sqrt[4]{R_0^4 + 2t}$  for  $R_0 > 0$ . In Figure 4.1 we plot the evolution of the error  $err(h) = \|\mathcal{I}_h x(t) - x_h(t)\|_{L^2}$  in time. Thereby, the  $L^2$ -error is evaluated on the polygonal curve  $x_h(t)$  and  $\mathcal{I}_h$  denotes the nodal interpolation of  $x(t)$  at the projected positions of the nodes of  $x_h(t)$  in direction  $\nabla\gamma(n[x_h(t)])$ . In Table 4.1 and 4.2 we provide results on the experimental order of convergence  $eoc = \log(err(h_1)/err(h_2))/\log(h_1/h_2)$ , with  $h_1 > h_2$ , for varying grid and

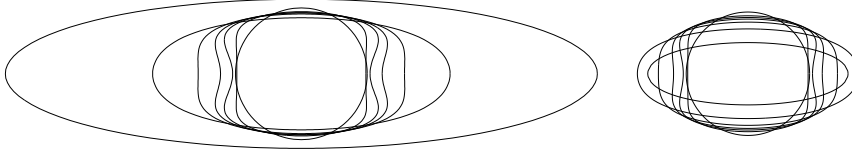


Figure 4.4.: The impact of the parameter  $\lambda$  is shown for the evolution of a circle to an ellipse with aspect ratio 4 : 1 (i.e.  $a_1 = 4$  and  $a_2 = 1$ ). We evolve polygons with 160 vertices approximating the unit sphere as initial curve,  $h_0 = 0.0393$  and  $\tau = \tilde{\tau} = 0.01$ ,  $h = 0.000393$ . On the left  $\lambda = 0.025$  and on the right  $\lambda = 4$ .

time step size for the evolution of the circle and the ellipse.

Now, we want to study crystalline anisotropies as considered in [103] for  $z \in \mathbb{R}^2$

$$\gamma(z) = \|z\|_{\ell^1} = |z_1| + |z_2|$$

and

$$\gamma(z) = \|z\|_{\ell^\infty} = \max\{|z_1|, |z_2|\} = \frac{|z_1 + z_2|}{2} + \frac{|z_1 - z_2|}{2}.$$

As already pointed out, even though the formulation of the scheme itself doesn't explicitly need assumptions on the smoothness of  $\gamma$ , the application of Newton method (i.e. the optimization algorithm) requires the computation of derivatives of  $\gamma$  up to order 3. In fact, we use the following regularization: For a small parameter  $\varepsilon > 0$  we regularize the  $\ell^1$ -norm by

$$\ell_\varepsilon^1(z) = \sum_{l=1}^2 \sqrt{\varepsilon|z|^2 + z_l^2}.$$

Since in  $\mathbb{R}^2$  the  $\ell^\infty$ -norm equals a rotated and scaled  $\ell^1$ -norm we use as regularization of the  $\ell^\infty$ -norm

$$\ell_\varepsilon^\infty(z) = \frac{\sqrt{\varepsilon|z|^2 + (z_1 + z_2)^2}}{2} + \frac{\sqrt{\varepsilon|z|^2 + (z_1 - z_2)^2}}{2}.$$

Figure 4.2 shows the evolution of a sphere with respect to the regularized  $\ell^\infty$ -norm under the associated anisotropic Willmore flow with anisotropy  $\gamma(\cdot) = \|\cdot\|_{\ell_\varepsilon^1}$  for  $\varepsilon = 0.0001$ . Results on the self-similar evolution of spheres with respect to the regularized  $\ell^1$ -norm are depicted in Figure 4.3. In these simulations, we use the analogous regularization for the dual anisotropy  $\gamma^*$  required in the algorithm.

Next, we generalize Willmore flow and replace the Willmore energy by the modified energy

$$\int_{\mathcal{M}[x]} \left( \frac{1}{2} \mathbf{h}_\gamma^2 + \lambda \right) \gamma(n[x]) da,$$

with a second term given by the anisotropic area weighted with a constant  $\lambda > 0$ . The incorporation of this generalized energy in our computational approach is straightforward. The generalized flow com-

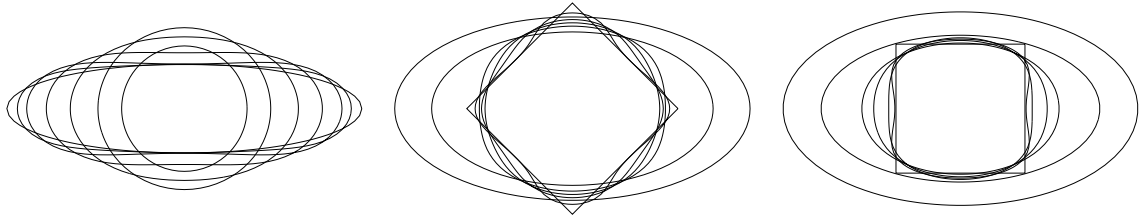


Figure 4.5.: The evolution of different initial shapes for different anisotropies is displayed. For all computations we use 100 vertices and choose  $\lambda = 0.25$ . On the left we start with an ellipse with aspect ratio 4 : 1 under an isotropic flow with  $\gamma(\cdot) = \|\cdot\|$  ( $h_0 = 0.1739$ ,  $\tau = h_0$ ,  $\tilde{\tau} = h_0^2$ ). Results are shown at  $t = 0, 0.1739, 0.5218, 1.739, 3.478, 6.956, 173.9$ . In the middle and on the right an ellipsoidal anisotropy with aspect ratio 2 : 1 is used (i.e.  $a_1 = 2$ ,  $a_2 = 1$ ). In the first case (middle), we take as initial shape the unit sphere for the  $l^1$ -norm ( $h_0 = 0.0566$ ,  $\tau = \tilde{\tau} = 0.001 h_0$ ) and results are displayed at  $t = 0, 0.00017, 0.00085, 0.00169, 0.006, 0.056, 0.251$ . In the second example (right), the initial shape is the unit sphere for the  $l^\infty$ -norm ( $h_0 = 0.08$  and  $\tau = \tilde{\tau} = 0.01 h_0$ ) and results are depicted for  $t = 0, 0.0024, 0.008, 0.04, 0.08, 0.8, 4.8$ .

bines expansive forcing with respect to the anisotropic Willmore flow of curves with contractive forcing due to the anisotropic mean curvature motion associated to the anisotropic area functional. Thus, for the generalized model we expect convergence to a limit shape given by a scaled Wulff shape, where the scaling depends on the factor  $\lambda$ . Figure 4.4 shows the impact of the factor  $\lambda$  on the evolution, whereas in Figure 4.5 we compare the evolution of different initial shapes under the generalized anisotropic Willmore flow for different anisotropies.

## 4.2. Spatial Discretization

In Chapter 2 and 3 we described the spatial discretization with Loop's subdivision surfaces to study partial differential equations on a given surface  $\mathcal{M}$ . Therefore, we introduced the Loop discretization space  $V_h$  in (3.4). Since we do not deal with scalar functions on a surface but with vector functions describing the surface itself we define the corresponding discretization space by

$$\mathcal{V}_h = \left\{ Y : \Omega \rightarrow \mathbb{R}^3 \mid Y_k \in V_h, k = 1, 2, 3 \right\}. \quad (4.15)$$

Therefore we will denote the basis functions  $i \in \mathcal{I}_v$  and coordinates  $k = 1, 2, 3$  of  $\mathcal{V}_h$  by  $\Psi_i^k$  where  $\Psi_i^1 = (\Psi_i, 0, 0)^T$ ,  $\Psi_i^2 = (0, \Psi_i, 0)^T$  and  $\Psi_i^3 = (0, 0, \Psi_i)^T$ .

For a given discrete initial surface  $\mathcal{M}_0$  represented by  $Y_0 = Y(0)$ , we want to compute the spatially discrete counterpart to (4.2) for a family of parameterizations  $\{Y(t)\}_{t>0}$  where  $Y(t) \in \mathcal{V}_h$  by solving

$$\mathbf{G}_Y(Y_t, \Psi) = -\mathbf{E}'[Y](\Psi) \quad \forall \Psi \in \mathcal{V}_h. \quad (4.16)$$

In the presence of constraints we look for a spatially discrete solution of (4.3) by solving

$$\mathbf{G}_Y(Y_t, \Psi) = -\mathbf{E}'[Y](\Psi) - \lambda \mathbf{C}'[Y](\Psi) \quad \forall \Psi \in \mathcal{V}_h, \quad (4.17a)$$

$$\mathbf{C}'[Y](Y_t) = 0, \quad (4.17b)$$

where  $\lambda = \lambda(t) \in \mathbb{R}$  denotes the discrete Lagrange multiplier. Because  $\text{span}_{i \in \mathcal{I}_v, k=1,2,3} \{\Psi_i^k\}$  are a basis of  $\mathcal{V}_h$  the problem reduces to finding a solution of

$$\mathbf{G}_Y(Y_t, \Psi_i^k) = -\mathbf{E}'[Y](\Psi_i^k) - \lambda \mathbf{C}'[Y](\Psi_i^k) \quad i \in \mathcal{I}_v, k = 1, 2, 3, \quad (4.18a)$$

$$\mathbf{C}'[Y](Y_t) = 0 \quad (4.18b)$$

and analogously for (4.16).

Now we can introduce the straightforward spatially discrete counterparts of (4.2) resp. (4.3). First, for a given discretization  $Y$  of a surface  $\mathcal{M}$ , we define the discrete energy functionals to (4.4) by

$$\mathbf{E}[Y] = \int_{\Omega} r(Y, DY, D^2Y) \, da = \sum_{k \in \mathcal{I}_c} \int_{\Delta} r(Y(\xi, k), DY(\xi, k), D^2Y(\xi, k)) \, d\xi. \quad (4.19)$$

For given functions  $\Psi, \Phi \in \mathcal{V}_h$  we define the discrete metric to (4.5) by

$$\begin{aligned} \mathbf{G}_Y^{\beta_0, \beta_1}(\Psi, \Phi) &= \int_{\mathcal{M}[Y]} \Psi \Phi + \beta_0 \nabla_{\mathcal{M}} \Psi : \nabla_{\mathcal{M}} \Phi + \beta_1 \Delta_{\mathcal{M}} \Psi \Delta_{\mathcal{M}} \Phi \, da \\ &= \sum_{k \in \mathcal{I}_c} \int_{\Delta} (\Psi \Phi + \beta_0 \nabla_{\mathcal{M}} \Psi : \nabla_{\mathcal{M}} \Phi + \beta_1 \Delta_{\mathcal{M}} \Psi \Delta_{\mathcal{M}} \Phi) \circ Y(\xi, k) \sqrt{\det G(\xi, k)} \, d\xi. \end{aligned} \quad (4.20)$$

Accordingly for our test problems, we define the discrete counterpart of (4.4) by

$$\begin{aligned} \mathbf{E}^{\alpha_0, \alpha_1}[Y] &= \int_{\mathcal{M}[Y]} \alpha_0 + \frac{\alpha_1}{2} \mathbf{h}^2 \, da \\ &= \sum_{k \in \mathcal{I}_c} \int_{\Delta} \left( \alpha_0 + \frac{\alpha_1}{2} (\mathbf{h}(\xi, k))^2 \right) \sqrt{\det G(\xi, k)} \, d\xi \end{aligned} \quad (4.21)$$

and analogously the constraint (4.8) is discretized by

$$\mathbf{C}[Y] = \frac{1}{3} \sum_{k \in \mathcal{I}_c} \int_{\Delta} N(\xi, k) \cdot Y(\xi, k) \sqrt{\det G(\xi, k)} \, d\xi - V_0. \quad (4.22)$$

The evaluation of (4.20), (4.21) and (4.22) is then performed by quadrature as discussed in Chapter 3 but here we will always apply the full Gaussian quadrature of the corresponding regular element.

### 4.3. Higher Order Implicit Runge–Kutta Methods in Time

In this section we describe the higher order time discretization of (4.2) and (4.3) with implicit Runge–Kutta methods. First, we give a brief review of the time integration of ordinary differential equations via implicit Runge–Kutta methods. For a more comprehensive overview on this topic, we recommend [73, 76] and [75]. Second, we apply implicit Runge–Kutta schemes to our discrete gradient flow setups (4.16) and (4.17).

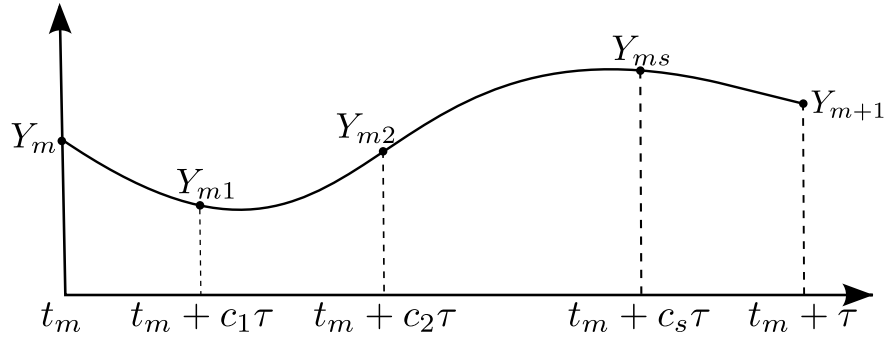


Figure 4.6.: Implicit Runge-Kutta time discretization: For a given Butcher tableau we compute the solution  $Y_{m+1}$  at the new time step  $t_m + \tau$  by solving the gradient flow for the internal stages  $Y_{mi}$ .

**Implicit Runge–Kutta methods for ODEs** Let us consider the following first-order *ordinary differential equation* in standard form

$$\begin{aligned} Y_t &= F(t, Y) \\ Y_0 &= Y(0) \end{aligned}$$

where  $F : [0, T] \times \mathcal{V}_h \rightarrow \mathbb{R}^{|\mathcal{I}_v|}$  and  $Y(0)$  is given. To determine the approximations  $Y_1, \dots, Y_n$  of  $Y(t_1), \dots, Y(t_n)$  for a given time-equidistant grid  $0 = t_0 < t_1 < \dots < t_n \leq T$  with step size  $\tau$  we first compute the internal stages  $Y_{m1}, \dots, Y_{ms}$  (see Figure 4.6) via

$$Y_{mi} = Y_m + \tau \sum_{j=1}^s a_{ij} F(t_m + c_i\tau, Y_{mj}) \quad i = 1, \dots, s,$$

then the approximation at time  $t_{m+1} = t_m + \tau$  is found by setting

$$Y_{m+1} = Y_m + \tau \sum_{i=1}^s b_i F(t_m + c_i\tau, Y_{mi}).$$

We note that these two steps are carried out at each time step  $t_1, t_2, \dots, t_n$ . In general, the first step requires solving a nonlinear system of equations, which is usually done by using Newton’s method.

An  $s$ -stage implicit Runge–Kutta method is uniquely defined by its coefficients  $A = (a_{ij})$ ,  $b^T = (b_1, \dots, b_s)$  and  $c^T = (c_1, \dots, c_s)$ . It is typically represented by the so-called *Butcher tableau*

$c_1$	$a_{11}$	$a_{12}$	$\dots$	$a_{1s}$
$c_2$	$a_{21}$	$a_{22}$	$\dots$	$a_{2s}$
$\vdots$	$\vdots$		$\ddots$	$\vdots$
$c_s$	$a_{s1}$	$a_{s2}$	$\dots$	$a_{ss}$
	$b_1$	$b_2$	$\dots$	$b_s$

#### 4. Parametric Gradient Flows

---

We say that the method is of *stage order*  $r \geq 1$  if this is the largest integer such that

$$\max_{0 \leq m \leq n} \max_{1 \leq i \leq s} \|Y_{mi} - Y(t_m + c_i \tau)\| = \mathcal{O}(\tau^r).$$

The *classical order*  $q \geq r$  is the largest integer such that

$$\max_{0 \leq m \leq n} \|Y_m - Y(t_m)\| = \mathcal{O}(\tau^q).$$

It is important to note that the terms  $\mathcal{O}(\tau^r)$  and  $\mathcal{O}(\tau^q)$  do not only depend on the coefficients of the Runge–Kutta method but also on the regularity of the right-hand side  $F$ .

**Implicit Runge-Kutta for gradient flows** Now, we want to solve for given surface  $\mathcal{M}_0 = \mathcal{M}[Y_0]$

$$\mathbf{G}_Y(\dot{Y}, \Psi) = -\mathbf{E}'[Y](\Psi) - \Lambda \mathbf{C}'[Y](\Psi) \quad \forall \Psi \in \mathcal{V}_h, \quad (4.23a)$$

$$\mathbf{C}'[Y](\dot{Y}) = 0, \quad (4.23b)$$

where  $\mathbf{E}$  and  $\mathbf{C}$  are twice continuously differentiable,  $\Lambda = \Lambda(t) \in \mathbb{R}$  denotes the Lagrange multiplier and  $\dot{\cdot} = \frac{\partial}{\partial t}$  denotes the partial derivative with respect to time  $t$  for notational reasons. Then, along every exact solution we have

$$\frac{d}{dt} \mathbf{E}[Y(t)] = \mathbf{E}'[Y(t)](\dot{Y}(t)) = \mathbf{E}'[Y(t)](\dot{Y}(t)) + \Lambda \mathbf{C}'[Y](\dot{Y}(t)) = -\mathbf{G}_Y(\dot{Y}(t), \dot{Y}(t)) \leq 0$$

implying that  $\mathbf{E}[Y(t)]$  is monotonically decreasing.

**Method formulation** For the numerical integration of the ordinary differential equation (4.23), we consider an  $s$ -stage implicit Runge–Kutta method with time-step size  $\tau = t_{n+1} - t_n$ , given by

$$Y_{m+1} = Y_m + \tau \sum_{i=1}^s b_i \dot{Y}_{mi}, \quad (4.24a)$$

$$Y_{mi} = Y_m + \tau \sum_{j=1}^s a_{ij} \dot{Y}_{mj}, \quad i = 1, \dots, s, \quad (4.24b)$$

where the internal stages satisfy

$$\mathbf{G}_{Y_{mi}}(\dot{Y}_{mi}, \Psi) = -\mathbf{E}'[Y_{mi}](\Psi) - \Lambda_i \mathbf{C}'[Y_{mi}](\Psi) \quad \forall \Psi \in \mathcal{V}_h, \quad i = 1, \dots, s, \quad (4.24c)$$

$$\mathbf{C}'[Y_{mi}](\dot{Y}_{mi}) = 0. \quad (4.24d)$$

**Gauss methods** Here, the  $\{b_i\}_{i=1}^s$  are the weights of the  $s$ -stage Gaussian-quadrature and the  $\{c_i\}_{i=1}^s$  are the nodes of this quadrature transformed to the interval  $[0, 1]$ . The coefficients of the matrix  $A$  are determined from the conditions

$$\sum_{j=1}^s a_{ij} c_j^{k-1} = \frac{c_i^k}{k}, \quad i, k = 1, \dots, s.$$

It is known that the Gauss–Runge–Kutta method (GRK) is of stage order  $r = s$  and classical order  $q = 2s$ .

The lowest order method of this class is the implicit midpoint rule. For the sake of completeness, we list the Gauss–Runge–Kutta methods of classical order up to 6 in Table 4.3.

$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2} - \frac{\sqrt{3}}{6}$	$\frac{1}{4}$	$\frac{1}{4} - \frac{\sqrt{3}}{6}$	$\frac{1}{2} - \frac{\sqrt{15}}{10}$	$\frac{5}{36}$	$\frac{2}{9} - \frac{\sqrt{15}}{15}$	$\frac{5}{36} - \frac{\sqrt{15}}{30}$
$\frac{1}{2} + \frac{\sqrt{3}}{6}$	$\frac{1}{4} + \frac{\sqrt{3}}{6}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2} + \frac{\sqrt{15}}{10}$	$\frac{5}{36} + \frac{\sqrt{15}}{24}$	$\frac{2}{9}$	$\frac{5}{36} - \frac{\sqrt{15}}{24}$	$\frac{5}{36}$
$1$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{5}{18}$	$\frac{4}{9}$	$\frac{5}{18}$	$\frac{4}{9}$	$\frac{5}{18}$	$\frac{5}{18}$
(a)	(b)	(c)						

Table 4.3.: Butcher tables of different Gauss-Runge-Kutta methods: (a) 1-stage 2nd order Gauss method (Implicit Midpoint Rule), (b) 2-stage 4th order Gauss method and (c) 3-stage 6th order Gauss method.

**RadauIIA methods** Analogously to the previous paragraph, the  $\{b_i\}_{i=1}^s$  are the weights of the  $s$ -stage Radau-quadrature and the  $\{c_i\}_{i=1}^s$  are the roots of the polynomial

$$\frac{d^{s-1}}{dx^{s-1}} (x^{s-1}(x-1)^s).$$

The coefficients of the matrix  $A$  are determined from the conditions

$$\sum_{j=1}^s a_{ij} c_j^{k-1} = \frac{c_i^k}{k} \quad i, k = 1, \dots, s.$$

It is known that the RadauIIA method is of stage order  $r = s$  and classical order  $q = 2s - 1$ .

The lowest order method of this class is the backward Euler method (implicit Euler). Here, we also list the RadauIIA methods up to classical order 5 in Table 4.4.

$1$	$1$	$\frac{1}{3}$	$\frac{5}{12}$	$-\frac{1}{12}$	$\frac{4-\sqrt{6}}{10}$	$\frac{88-7\sqrt{6}}{360}$	$\frac{296-169\sqrt{6}}{1800}$	$\frac{-2+3\sqrt{6}}{225}$
$1$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{4+\sqrt{6}}{10}$	$\frac{296+169\sqrt{6}}{1800}$	$\frac{88+7\sqrt{6}}{360}$	$\frac{-2-3\sqrt{6}}{225}$	$\frac{1}{9}$
$1$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{16-\sqrt{6}}{36}$	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$	$\frac{1}{9}$
(a)	(b)	(c)						

Table 4.4.: Butcher tableaux of different RadauIIA methods: (a) 1-stage 1st order RadauIIA method (Implicit Euler), (b) 2-stage 3rd order RadauIIA method and (c) 3-stage 5th order RadauIIA method.

## 4.4. Robustness and Energy Decay

Hairer and Lubich proved in [74] that under certain assumptions, implicit Runge–Kutta methods diminish the energy of gradient systems in every step. In this section, we adapt their theorem to our situation. Since we are dealing with a problem depending on the mesh size  $h$ , we will show which kind of time step size restriction is needed in order to fulfill the required assumptions guaranteeing the energy decay.

In the following we will differentiate between second and fourth-order problems indicated by the variable  $\ell$ . Here,  $\ell = 1$  refers to the second-order case, i.e.  $r(Y, DY, D^2Y) = r(Y, DY)$ , and  $\ell = 2$  to the fourth-order case.

**Assumptions.** We assume that

- (i) the Runge-Kutta method is algebraically stable, i.e.

$$b_i \geq 0, \quad i = 1, \dots, s,$$

$$(m_{ij})_{ij} = (b_i a_{ij} + b_j a_{ji} - b_i b_j)_{ij} \text{ is symmetric positive semi-definite}$$

It is known that the GRK method as well as the RadauIIA method are both algebraically stable.

- (ii) there exists a  $\delta > 0$  such that

$$\|Y_{m+1} - Y_m\|_{W^{\ell, \infty}} \leq \delta, \quad \|Y_{mi} - Y_m\|_{W^{\ell, \infty}} \leq \delta, \quad i = 1, \dots, s \quad \text{and} \quad \ell = 1, 2. \quad (4.25)$$

- (iii) there exists a  $C_* > 0$  such that

$$\|D_0^2 r\|_{\infty}, \|D_1^2 r\|_{\infty}, \|D_2^2 r\|_{\infty}, \|D_0 D_1 r\|_{\infty}, \|D_0 D_2 r\|_{\infty}, \|D_1 D_2 r\|_{\infty} \leq C_*,$$

where  $D_0$ ,  $D_1$  and  $D_2$  denote the partial derivative with respect to the first, second and third argument of  $r$ , respectively.

- (iv) the operator  $\mathcal{A} : (H^\ell)^s \times (H^\ell)^s \rightarrow \mathbb{R}$  defined by

$$\mathcal{A}(\mathbf{Z}, \mathbf{W}) = \sum_{i=1}^s b_i \mathbf{G}_{Y_{mi}}(Z_i, W_i) + \frac{\tau}{2} \sum_{i,j=1}^s m_{ij} \mathbf{E}''[Y_m](Z_i, W_j)$$

is positive definite, i.e.

$$\mathcal{A}(\mathbf{Z}, \mathbf{Z}) \geq \alpha \sum_{i=1}^s \|Z_i\|_{L^2}^2 \quad \text{with} \quad \alpha > 0.$$

We used the notation  $\mathbf{Z} = (Z_1, \dots, Z_s)^T$  and  $\mathbf{W} = (W_1, \dots, W_s)^T$ .

**Theorem 4.4.1.** Consider the system (4.24) with twice differentiable functions  $\mathbf{E}[Y]$  and  $\mathbf{C}[Y]$ . If the above assumptions (i)-(iv) are satisfied and if  $\tau \delta h^{-2\ell}$  is bounded by a sufficiently small constant, then we have

$$\mathbf{E}[Y_{m+1}] \leq \mathbf{E}[Y_m].$$



*Proof.* The proof uses similar arguments as in the proof of Theorem 2.1 in [74]. We define  $p(t) = \mathbf{E}[Y_m + t(Y_{m+1} - Y_m)]$ , then Taylor expansion yields

$$p(1) = p(0) + p'(0) + \frac{1}{2}p''(\rho) \quad \text{where } \rho \in [0, 1].$$

Thus we have

$$\mathbf{E}[Y_{m+1}] = \mathbf{E}[Y_m] + \mathbf{E}'[Y_m](Y_{m+1} - Y_m) + \frac{1}{2}\mathbf{E}''[Y_m + \xi(Y_{m+1} - Y_m)](Y_{m+1} - Y_m, Y_{m+1} - Y_m). \quad (4.26)$$

On the other hand, we again use the Taylor expansion to get for all  $i = 1, \dots, s$

$$\mathbf{E}'[Y_{mi}](Y_{m+1} - Y_m) = \mathbf{E}'[Y_m](Y_{m+1} - Y_m) + \int_0^1 \mathbf{E}''[Y_m + \theta(Y_{m+1} - Y_m)](Y_{mi} - Y_m, Y_{m+1} - Y_m)d\theta.$$

Using the Runge–Kutta relation (4.24) to express the terms  $Y_{m+1} - Y_m$  and  $Y_{mi} - Y_m$ , extracting  $\mathbf{E}'[Y_m](Y_{m+1} - Y_m)$  from the last equation and inserting it into (4.26), it follows that

$$\begin{aligned} \mathbf{E}[Y_{m+1}] &= \mathbf{E}[Y_m] + \tau \sum_{i=1}^s b_i (\mathbf{E}'[Y_{mi}](\dot{Y}_{mi}) + \Lambda_i \mathbf{C}'[Y_{mi}](\dot{Y}_{mi})) \\ &\quad + \frac{1}{2}\tau^2 \sum_{i,j=1}^s b_i b_j \mathbf{E}''[Y_m + \xi(Y_{m+1} - Y_m)](\dot{Y}_{mi}, \dot{Y}_{mj}) \\ &\quad - \tau^2 \sum_{i,j=1}^s b_i a_{ij} \int_0^1 \mathbf{E}''[Y_m + \theta(Y_{mi} - Y_m)](\dot{Y}_{mi}, \dot{Y}_{mj})d\theta \\ &= \mathbf{E}[Y_m] - \tau \sum_{i=1}^s b_i \mathbf{G}_{Y_{mi}}(\dot{Y}_{mi}, \dot{Y}_{mi}) + \frac{\tau^2}{2} \sum_{i,j=1}^s b_i b_j \mathbf{E}''[Y_m + \xi(Y_{m+1} - Y_m)](\dot{Y}_{mi}, \dot{Y}_{mj}) \\ &\quad - \frac{\tau^2}{2} \sum_{i,j=1}^s b_i a_{ij} \int_0^1 \mathbf{E}''[Y_m + \theta(Y_{mi} - Y_m)](\dot{Y}_{mi}, \dot{Y}_{mj})d\theta \\ &\quad - \frac{\tau^2}{2} \sum_{i,j=1}^s b_j a_{ji} \int_0^1 \mathbf{E}''[Y_m + \theta(Y_{mj} - Y_m)](\dot{Y}_{mj}, \dot{Y}_{mi})d\theta \\ &= \mathbf{E}[Y_m] - \tau \sum_{i=1}^s b_i \mathbf{G}_{Y_{mi}}(\dot{Y}_{mi}, \dot{Y}_{mi}) - \frac{\tau^2}{2} \left( \sum_{i,j=1}^s m_{ij} \mathbf{E}''[Y_m](\dot{Y}_{mi}, \dot{Y}_{mj}) + R_{ij} \right). \\ &= \mathbf{E}[Y_m] - \tau \left( \mathcal{A}(\dot{\mathbf{Y}}_m, \dot{\mathbf{Y}}_m) + \frac{\tau}{2} \mathcal{R}(\dot{\mathbf{Y}}_m, \dot{\mathbf{Y}}_m) \right) \end{aligned}$$

where

$$\mathcal{R}(\dot{\mathbf{Y}}_m, \dot{\mathbf{Y}}_m) = \sum_{i,j=1}^s 2b_i a_{ij} \int_0^1 \mathbf{E}''[Y_m + \theta(Y_{mi} - Y_m)](\dot{Y}_{mi}, \dot{Y}_{mj}) - \mathbf{E}''[Y_m](\dot{Y}_{mi}, \dot{Y}_{mj})d\theta$$

$$- \sum_{i,j=1}^s b_i b_j \left( \mathbf{E}''[Y_m + \xi(Y_{m+1} - Y_m)](\dot{Y}_{mi}, \dot{Y}_{mj}) - \mathbf{E}''[Y_m](\dot{Y}_{mi}, \dot{Y}_{mj}) \right)$$

and  $\dot{\mathbf{Y}}_m = (\dot{Y}_{m1}, \dots, \dot{Y}_{ms})^T$ . By assumption (iv) we have  $\mathcal{A}(\dot{\mathbf{Y}}_m, \dot{\mathbf{Y}}_m) \geq \alpha \sum_{i=1}^s \|\dot{Y}_{mi}\|_{L^2}^2$ . Therefore, our problem reduces to show that

$$\tau \mathcal{R}(\dot{\mathbf{Y}}_m, \dot{\mathbf{Y}}_m) \leq c_\alpha \sum_{i=1}^s \|\dot{Y}_{mi}\|_{L^2}^2,$$

with  $c_\alpha \leq \alpha$ .

For  $U, \Psi, \Upsilon \in \mathcal{V}_h$  we first observe that

$$\begin{aligned} \mathbf{E}'[U](\Psi) &= \int_{\Omega} D_0 r(U, DU, D^2 U)(\Psi) + D_1 r(U, DU, D^2 U)(D\Psi) + D_2 r(U, DU, D^2 U)(D^2 \Psi) \, d\xi, \\ \mathbf{E}''[U](\Psi, \Upsilon) &= \int_{\Omega} D_0^2 r(U, DU, D^2 U)(\Psi, \Upsilon) + D_1^2 r(U, DU, D^2 U)(D\Psi, D\Upsilon) \\ &\quad + D_2^2 r(U, DU, D^2 U)(D^2 \Psi, D^2 \Upsilon) + D_1 D_0 r(U, DU, D^2 U)(\Psi, D\Upsilon) \\ &\quad + D_2 D_0 r(U, DU, D^2 U)(\Psi, D^2 \Upsilon) + D_0 D_1 r(U, DU, D^2 U)(D\Psi, \Upsilon) \\ &\quad + D_0 D_2 r(U, DU, D^2 U)(D^2 \Psi, \Upsilon) + D_2 D_1 r(U, DU, D^2 U)(D\Psi, D^2 \Upsilon) \\ &\quad + D_1 D_2 r(U, DU, D^2 U)(D^2 \Psi, D\Upsilon) \, d\xi. \end{aligned}$$

Thus, by assumption (iii), Cauchy-Schwarz inequality, Young's inequality and inverse inequality, it follows that for all  $U, \tilde{U}, \Psi, \Upsilon \in \mathcal{V}_h$

$$\begin{aligned} |\mathbf{E}''[U](\Psi, \Upsilon) - \mathbf{E}''[\tilde{U}](\Psi, \Upsilon)| &\leq C_* \|U - \tilde{U}\|_{W^{\ell, \infty}} \left( \sum_{i=0}^{\ell} \|D^i \Psi\|_{L^2}^2 + \|D^i \Upsilon\|_{L^2}^2 \right) \\ &\leq C_* h^{-2\ell} \|U - \tilde{U}\|_{W^{\ell, \infty}} (\|\Psi\|_{L^2}^2 + \|\Upsilon\|_{L^2}^2). \end{aligned}$$

This relation together with assumption (ii) yield

$$\mathcal{R}(\dot{\mathbf{Y}}_m, \dot{\mathbf{Y}}_m) \leq C_0 \delta h^{-2\ell} \sum_{i=1}^s \|\dot{Y}_{mi}\|_{L^2}^2,$$

where  $C_0$  is independent of  $h$  and  $\tau$  but depends on  $C_*$  and the Runge–Kutta coefficients. Thus the assumption that  $\tau \delta h^{-2\ell}$  is sufficiently small completes the proof.  $\square$

**Time-step restriction** We now illustrate that under a mild time step size restriction, assumptions (i)-(iv) can be fulfilled.

- (i) Independently of the time step size, it is known that the GRK method as well as the RadauIIA method are both algebraically stable. In particular  $m_{ij} = 0$  for all GRK methods.
- (ii) •  $\ell = 1$  (e.g. MCF): We expect that  $\|y(t_m) - Y_m\|_{W^{1, \infty}} \leq C(\tau^q + h^{\tilde{p}})$  with  $\tilde{p} > 0$ , we get

$$\|Y_{m+1} - Y_m\|_{W^{1, \infty}} \leq \|Y_{m+1} - y(t_{m+1})\|_{W^{1, \infty}} + \|y(t_{m+1}) - y(t_m)\|_{W^{1, \infty}} + \|y(t_m) - Y_m\|_{W^{1, \infty}}$$

$$\leq C\tau + Ch^{\tilde{p}}.$$

Thus it follows that in this case  $\delta = \mathcal{O}(\tau + h^{\tilde{p}})$  which means that we have to have

$$\tau = \mathcal{O}(h^{\max(1, 2-\tilde{p})})$$

in order to satisfy the assumption that  $\tau\delta h^{-2}$  is sufficiently small. For Loop subdivision we expect that  $\tilde{p} = 2$  which results in  $\tau = \mathcal{O}(h)$  (see the theoretical results obtained by Arden [5] or our experimental study in Chapter 3).

- $\ell = 2$  (e.g. Willmore flow): We expect that  $\|y(t_m) - Y_m\|_{W^{2,\infty}} \leq C(\tau^q + h^{\hat{p}})$  with  $\hat{p} > 0$ , we get

$$\|Y_{m+1} - Y_m\|_{W^{2,\infty}} \leq C(\tau + h^{\hat{p}}).$$

Thus, for  $\ell = 2$ , we expect that  $\delta = \mathcal{O}(\tau + h^{\hat{p}})$ . Therefore, we need a stronger time step size restriction, namely

$$\tau = \mathcal{O}(h^{\max(2, 4-\hat{p})}),$$

in order to satisfy the assumption that  $\tau\delta h^{-4}$  is sufficiently small. For arbitrary valences a Loop subdivision surface is only in  $W^{2,p^*}$  with  $p^* < \infty$  (see Chapter 2 or [117]) and only purely regular meshes (i.e. a torus) this requirement is fulfilled and we expect that  $\hat{p} = 2$  (see [86]) such that  $\tau = \mathcal{O}(h^2)$ . In all other cases we expect that the time step size depends much stronger on the spatial grid size.

- (iii) Since we consider only algebraically stable Runge–Kutta methods (i.e.  $b_i > 0$ ), and we expect that there exists a constant  $\gamma > 0$  such that  $\mathbf{G}_{Y_{mi}}(\varphi, \varphi) \geq \gamma \mathbf{G}_{Y_m}(\varphi, \varphi)$  for all  $i = 1, \dots, s$ , we observe

$$\begin{aligned} \mathcal{A}(\mathbf{Z}, \mathbf{Z}) &= \sum_{i=1}^s b_i \mathbf{G}_{Y_{mi}}(Z_i, Z_i) + \frac{\tau}{2} \sum_{i,j=1}^s m_{ij} \mathbf{E}''[Y_m](Z_i, Z_j) \\ &\geq C\gamma \sum_{i=1}^s \|Z_i\|_{L^2}^2 + \frac{\tau}{2} \sum_{i,j=1}^s m_{ij} \mathbf{E}''[Y_m](Z_i, Z_j). \end{aligned}$$

We now study the following two cases.

- Case 1:  $m_{ij} = 0$  or  $\mathbf{E}''[Y_m](\cdot, \cdot)$  is positive semi-definite. Then it is clear, thanks to the algebraic stability of the method, assumption (iv) is fulfilled. Thus, in this case, we don't require any extra restriction.
- Case 2:  $m_{ij} \neq 0$  and  $\mathbf{E}''[Y_m](\cdot, \cdot)$  is not positive semi-definite. This is the worst case that can happen in term of time step size restriction. Then we will have to use the inverse inequality in order to bound the last term

$$\frac{\tau}{2} \sum_{i,j=1}^s m_{ij} \mathbf{E}''[Y_m](Z_i, Z_j) \leq C\tau h^{-2\ell} \sum_{i=1}^s \|Z_i\|_{L^2}^2,$$

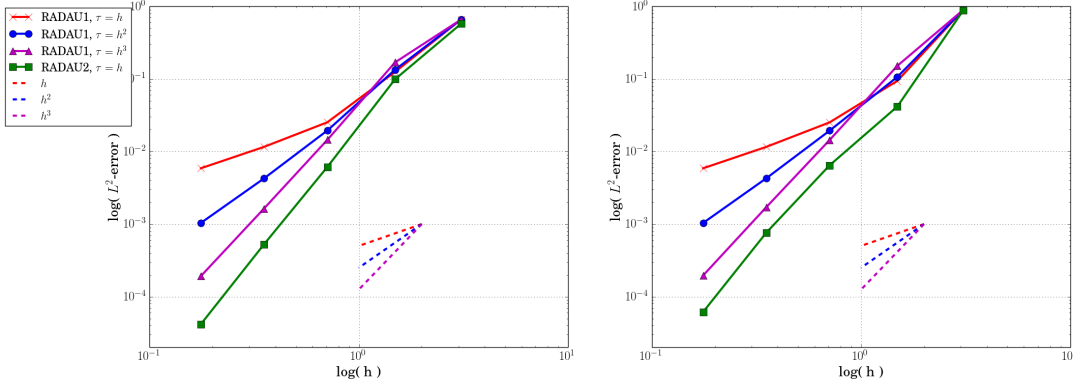


Figure 4.7.: Experimental order of convergence study for the spatial discretization of problem **(ex.1)**, left and **(ex.2)**, right. As initial approximation of a sphere with radius  $r_0 = 2.0$  we used a triangulation with  $|\mathcal{I}_v| = 18$ . For every refinement step we solve the corresponding interpolation problem to approximate the initial sphere. The error is measured in  $L^2$  at time  $t = 1.0$ .

which means that we need to have a time step size restriction

$$\tau = \mathcal{O}(h^{2\ell}).$$

**Conclusion.** For second-order problem ( $\ell = 1$ , e.g. MCF), we have a time step size restriction  $\tau = \mathcal{O}(h)$ , and in the worst case  $\tau = \mathcal{O}(h^2)$ . For fourth-order problem ( $\ell = 2$ , e.g. Willmore flow), in the best-case scenario the time step size scales like  $\mathcal{O}(h^p)$ , where  $p \in [2, 4)$ , and in the worst case  $\tau = \mathcal{O}(h^4)$ .

## 4.5. Algorithm

In this section we describe how to solve the system of equations (4.16) and (4.17). Here, we consider only the case (4.18) and the non-constrained problem follows analogously.

First, instead of considering the stage shapes  $Y_{mi}$  we consider  $\dot{Y}_{mi}$  as unknowns. For a given  $s$ -stage Runge-Kutta method and a time step  $m$  with parameterization  $Y_m$  we compute the next time step  $Y_{m+1}$  by first computing the roots of the following objective function

$$\mathbf{F}[\dot{\mathbf{Y}}, \mathbf{\Lambda}] = (F_1[\dot{\mathbf{Y}}, \mathbf{\Lambda}] \quad \dots \quad F_s[\dot{\mathbf{Y}}, \mathbf{\Lambda}] \quad F_{s+1}[\dot{\mathbf{Y}}, \mathbf{\Lambda}] \quad \dots \quad F_{2s}[\dot{\mathbf{Y}}, \mathbf{\Lambda}])^T,$$

where

$$\begin{aligned} \mathbf{F}_i[\dot{\mathbf{Y}}, \mathbf{\Lambda}] &= \left( \mathbf{G}_{Y_{mi}}(\dot{Y}_{mi}, \Psi_l^\kappa) + \mathbf{E}'[Y_{mi}](\Psi_l^\kappa) + \Lambda_i \mathbf{C}'[Y_{mi}](\Psi_l^\kappa) \right)_{l \in \mathcal{I}_v, \kappa=1,2,3}, \\ \mathbf{F}_{s+i}[\dot{\mathbf{Y}}, \mathbf{\Lambda}] &= \mathbf{C}'[Y_{mi}](\dot{Y}_{mi}), \end{aligned}$$

where

$$\dot{\mathbf{Y}} = (\dot{Y}_{m1}, \dots, \dot{Y}_{ms}) \in \mathcal{V}_h \times \dots \times \mathcal{V}_h,$$

$$\mathbf{\Lambda} = (\Lambda_1, \dots, \Lambda_s) \in \mathbb{R}^s$$

and

$$Y_{mi} = Y_m + \tau \sum_{j=1}^s a_{ij} \dot{Y}_{mj}$$

for  $i = 1, \dots, s$ . This is done by solving the full Newton method where the Jacobian of  $\mathbf{F}[\dot{\mathbf{Y}}, \mathbf{\Lambda}]$  has the structure

$$D\mathbf{F}[\dot{\mathbf{Y}}, \mathbf{\Lambda}] = \begin{pmatrix} \frac{\partial \mathbf{F}_1[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \dot{Y}_{m1}} & \cdots & \frac{\partial \mathbf{F}_1[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \dot{Y}_{ms}} & \frac{\partial \mathbf{F}_1[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \Lambda_1} & \cdots & \frac{\partial \mathbf{F}_1[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \Lambda_s} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{F}_s[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \dot{Y}_{m1}} & \cdots & \frac{\partial \mathbf{F}_s[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \dot{Y}_{ms}} & \frac{\partial \mathbf{F}_s[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \Lambda_1} & \cdots & \frac{\partial \mathbf{F}_s[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \Lambda_s} \\ \frac{\partial \mathbf{F}_{s+1}[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \dot{Y}_{m1}} & \cdots & \frac{\partial \mathbf{F}_{s+1}[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \dot{Y}_{ms}} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{F}_{2s}[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \dot{Y}_{ms}} & \cdots & \frac{\partial \mathbf{F}_{2s}[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \dot{Y}_{ms}} & 0 & \cdots & 0 \end{pmatrix}$$

with

$$\begin{aligned} \frac{\partial \mathbf{F}_i[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \dot{Y}_{mj}} &= \delta_{ij} \left( \mathbf{G}_{Y_{mi}}(\Psi_l^\kappa, \Psi_k^\eta) + a_{ij} \mathbf{G}_{Y_{mi}, Y}(\dot{Y}_{mi}, \Psi_l^\kappa)(\Psi_k^\eta) \right. \\ &\quad \left. + a_{ij} \mathbf{E}''[Y_{mi}](\Psi_l^\kappa, \Psi_k^\eta) + a_{ij} \Lambda_i \mathbf{C}''[Y_{mi}](\Psi_l^\kappa, \Psi_k^\eta) \right)_{l, k \in \mathcal{I}_v, \kappa, \eta = 1, 2, 3}, \\ \frac{\partial \mathbf{F}_{s+i}[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \dot{Y}_{mj}} &= \left( a_{ij} \mathbf{C}''[Y_{m1}](\dot{Y}_{m1}, \Psi_k^\eta) + \mathbf{C}'[Y_{m1}](\Psi_k^\eta) \right)_{k \in \mathcal{I}_v, \eta = 1, 2, 3}^T, \\ \frac{\partial \mathbf{F}_i[\dot{\mathbf{Y}}, \mathbf{\Lambda}]}{\partial \Lambda_j} &= \delta_{ij} \left( \mathbf{C}'[Y_{mi}](\Psi_l^\kappa) \right)_{l \in \mathcal{I}_v, \kappa = 1, 2, 3} \end{aligned}$$

for  $i = 1, \dots, s$ . For every Newton step  $\sigma \geq 0$  we solve the linear system

$$D\mathbf{F}[\dot{\mathbf{Y}}^\sigma, \mathbf{\Lambda}^\sigma] \cdot \Delta[\dot{\mathbf{Y}}^\sigma, \mathbf{\Lambda}^\sigma] = -\mathbf{F}[\dot{\mathbf{Y}}^\sigma, \mathbf{\Lambda}^\sigma]$$

and compute the next Newton step by

$$[\dot{\mathbf{Y}}^{\sigma+1}, \mathbf{\Lambda}^{\sigma+1}] = [\dot{\mathbf{Y}}^\sigma, \mathbf{\Lambda}^\sigma] + \eta \Delta[\dot{\mathbf{Y}}^\sigma, \mathbf{\Lambda}^\sigma],$$

where  $\eta \in [0, 1]$  is the step size computed by ([125], p.129). Finally, we compute the next time step

$$Y_{m+1} = Y_m + \tau \sum_{i=1}^s b_i \dot{Y}_{mi}.$$

For a detailed computation of the first and second variation of  $\mathbf{E}^{\alpha_0, \alpha_1}$ ,  $\mathbf{C}$  and  $\mathbf{G}_Y^{\beta_0, \beta_1}$  as well as the constraints see Appendix A.3.

### 4.6. Numerical Results

We have implemented the proposed method in C++ and performed various tests for different gradient flows, e.g. mean curvature flow and  $L^2$ -Willmore flow. We demonstrate the higher order consistency in space and in time through an experimental convergence study. Furthermore, we underline the robustness and stability of the proposed scheme with time steps up to the order of the spatial grid size  $h$  whereby the crucial discrete energy decay property is fulfilled. We will now refer to the  $s$ -stage RadauIIA method by RADAUs, i.e. RADAU1, RADAU2 and RADAU3, and for the  $s$ -stage Gauss–Runge–Kutta by GAUSSs, i.e. GAUSS1, GAUSS2 and GAUSS3.

At first, we numerically solve the evolution of the gradient flows with energy and metric

(ex.1)  $\mathbf{e} = \mathbf{e}^{1,0}$  and  $\mathbf{g}_y = \mathbf{g}_y^{0,0}$ ,

(ex.2)  $\mathbf{e} = \mathbf{e}^{1,1}$  and  $\mathbf{g}_y = \mathbf{g}_y^{0,0}$ .

Both gradient flows shrink spheres in time  $t$  with initial radius  $r_0$  under the law  $r(t) = \sqrt{r_0^2 - 4t}$  (see e.g. [107]). In Figure 4.7 we compare the continuous and the discrete solution of problem (ex.1) (on the left) and (ex.2) (on the right) and show the experimental order of convergence for the spatial

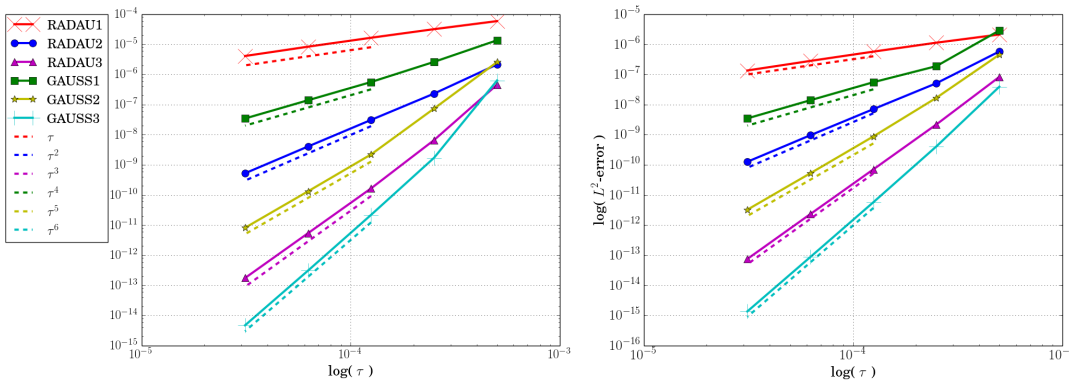


Figure 4.8.: Experimental order of convergence study for the time discretization of problem (ex.1), left and (ex.2), right. As initial approximation of a sphere with radius  $r_0 = 2.0$  we used a triangulation with  $|\mathcal{I}_v| = 1026$  and  $h = 0.27$ . The error is measured in  $L^2$  at time  $t = 0.8$ .

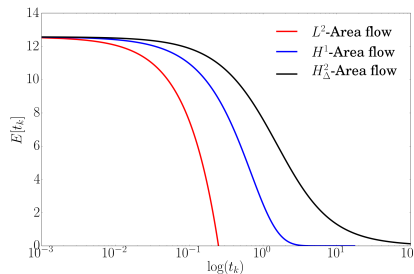


Figure 4.9.: Energy plot of  $\mathbf{E}^{1,0}$  for the shrinking of a sphere with different metrics  $L^2$ ,  $H^1$  and  $H^2_\Delta$ . The initial surface approximates a sphere of radius  $r_0 = 1$  with  $|\mathcal{I}_v| = 258$ ,  $h = 0.27$  and  $\mathbf{E}^{1,0}[Y_0] = 12.57$ . Here, we have chosen RADAU2 as time scheme with  $\tau = 1.e-3$ .

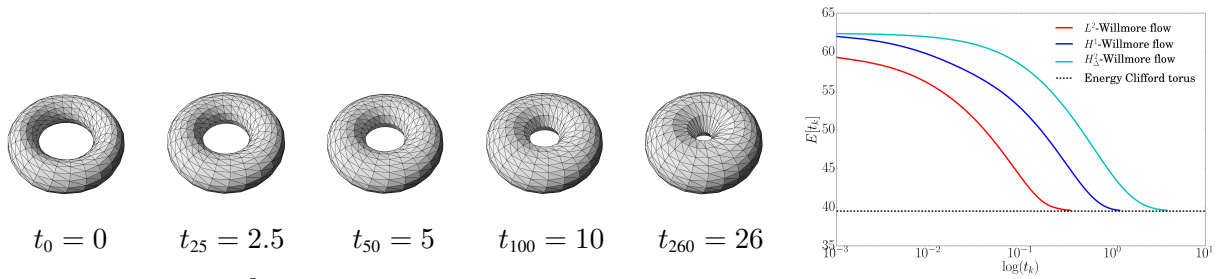


Figure 4.10.: Left:  $L^2$ -Willmore flow of torus towards the Clifford torus with  $|\mathcal{I}_v| = 384$ ,  $h = 0.6$  and  $\mathbf{E}^{0,1}[Y_0] = 62.39$  for the initial surface at  $t_0 = 0$ . Here, the time discretization scheme is RADAU2 with  $\tau = 0.1$  and depicted time steps are 0, 25, 50, 100 and 260. Right: Energy plot of  $\mathbf{E}^{0,1}[Y_m]$  for the corresponding  $L^2$ -,  $H^1$ - and  $H^2_{\Delta}$ -Willmore flow.

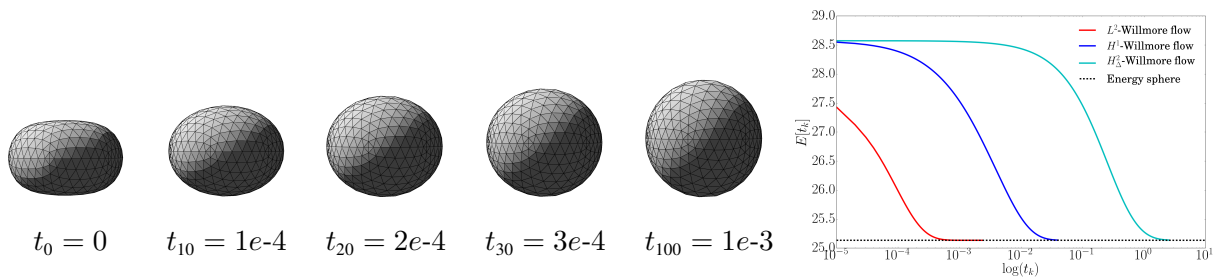


Figure 4.11.: Left:  $L^2$ -Willmore flow of an ellipsoid towards the sphere with  $|\mathcal{I}_v| = 450$ ,  $h = 7.48e-2$  and  $\mathbf{E}^{0,1}[Y_0] = 28.57$  for the initial surface at  $t_0 = 0$ . Here, the time discretization scheme is RADAU2 with  $\tau = 1.0e-5$  and depicted time steps are 0, 10, 20, 30 and 100. Right: Energy plot of  $\mathbf{E}^{0,1}[Y_m]$  for the corresponding  $L^2$ -,  $H^1$ - and  $H^2_{\Delta}$ -Willmore flow.

discretization measured in  $L^2$ . Here, the experimental behavior agrees with the suspected consistency error  $\mathcal{O}(\tau^q + h^p)$  with  $p = 3$  for Loop subdivision surfaces. In particular, for RADAU1 with time step  $\tau = h^s$  the experimental order of convergence scales like  $\mathcal{O}(h^s)$  for  $s = 1, 2, 3$ . In contrast, a time discretization with RADAU2 and  $\tau = h$  results in a consistency error  $\mathcal{O}(h^3)$  which allows much larger time steps with higher order consistency.

Now, we fix the spatial discretization, i.e.  $h$  is constant. In Figure 4.8 we depict an experimental convergence study of the higher order Runge-Kutta methods measured in  $L^2$  for the shrinking of a sphere, i.e. problems **(ex.1)** and **((ex.2))**. Here, we have chosen the spatial discretization such that the error is completely dominated by the time discretization error. Again, the results in Figure 4.8 agree with the suspected higher order consistency error  $\mathcal{O}(\tau^q)$ .

In Figure 4.9 we depict the energy plot for the evolution of the shrinking sphere of the area functional  $\mathbf{E}^{1,0}$  measured in  $L^2$ ,  $H^1$  and  $H^2_{\Delta}$ . For the  $L^2$  gradient flow we see a very fast evolution towards the singular configuration at time  $t = 0.25$  whereas the corresponding  $H^1$  and  $H^2$  gradient flows evolve much smoother in time. Furthermore, in Figure 4.10 and 4.11 we depict the energy plot for the evolution of a torus resp. a deformed sphere for the Willmore energy  $\mathbf{E}^{0,1}$  measured in  $L^2$ ,  $H^1$  and  $H^2_{\Delta}$ .

Now we investigate the evolution of described gradient flows for more complex examples. In Figure 4.12 and 4.13 we consider the evolution of a bunny surface towards a sphere with different gradient flows. In Figure 4.12 we depict three time steps of the  $L^2$ -Willmore flow. As described earlier by other authors [11, 20] the  $L^2$ -gradient can produce degenerate meshes for the evolution of complex surfaces.

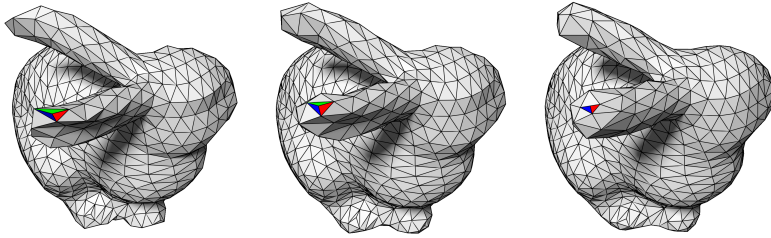


Figure 4.12.:  $L^2$ -Willmore flow of a complex surface towards a singular configuration with  $|\mathcal{I}_v| = 842$  and  $h = 0.12987$ . On the left we depict the initial mesh with  $t_0 = 0$ . In the middle the mesh after 10 time steps with  $\tau = 10^{-8}$ . After 60 time steps the green triangle degenerate for the mesh on the right.

In contrast, the evolution measured in metrics based on higher order derivatives (see Figure 4.13) does not form degenerate meshes. The resulting flows are stable and allow for large time steps.

Finally, the examples in Figure 4.14 and 4.15 underline again the robustness of the proposed method.



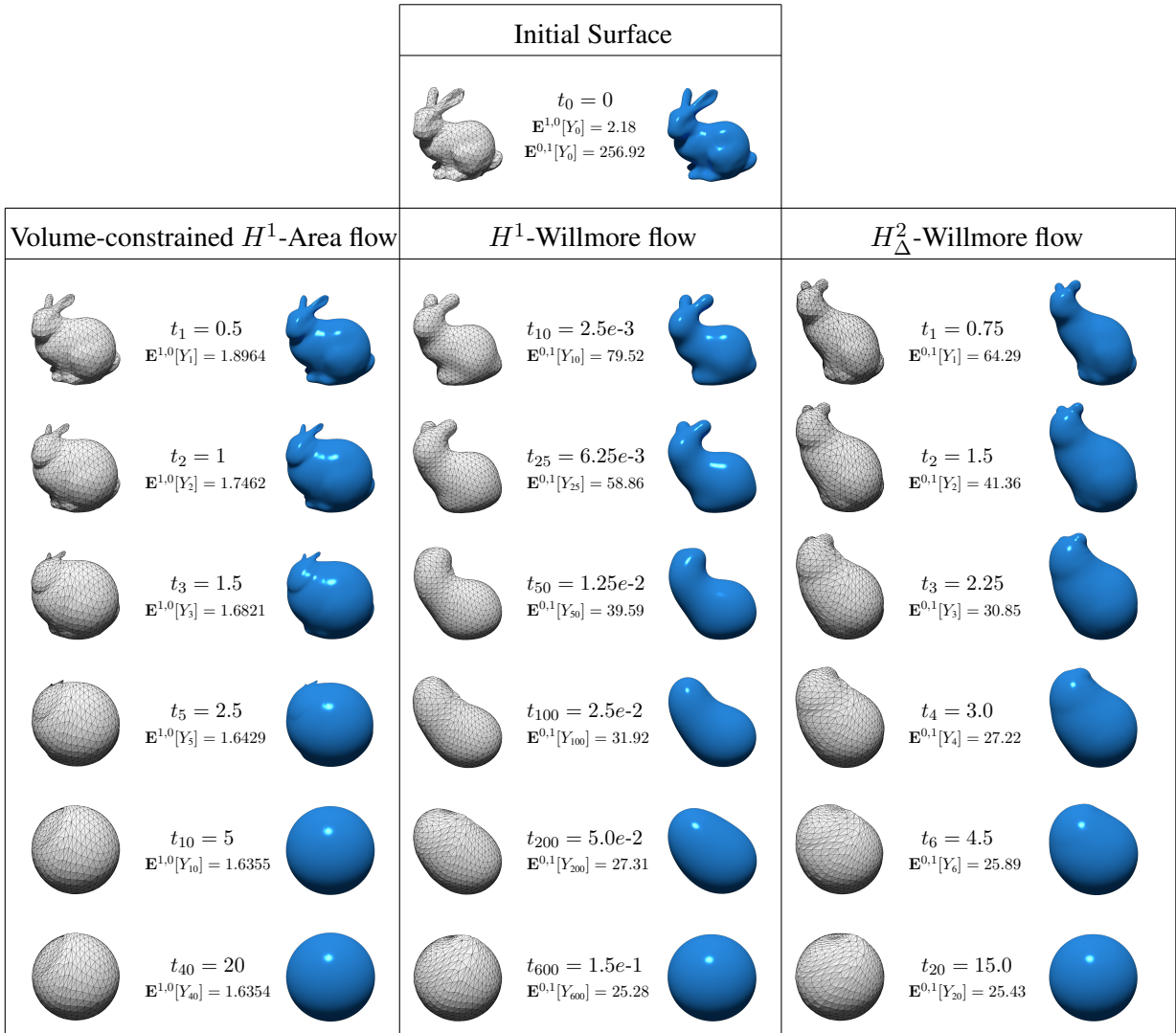


Figure 4.13.: Evolution of a bunny surface towards a sphere with different gradient flows. The initial surface is depicted at the top with  $|\mathcal{I}_v| = 842$  and  $h = 0.12987$  (left: coarse mesh, right: subdivision surface).

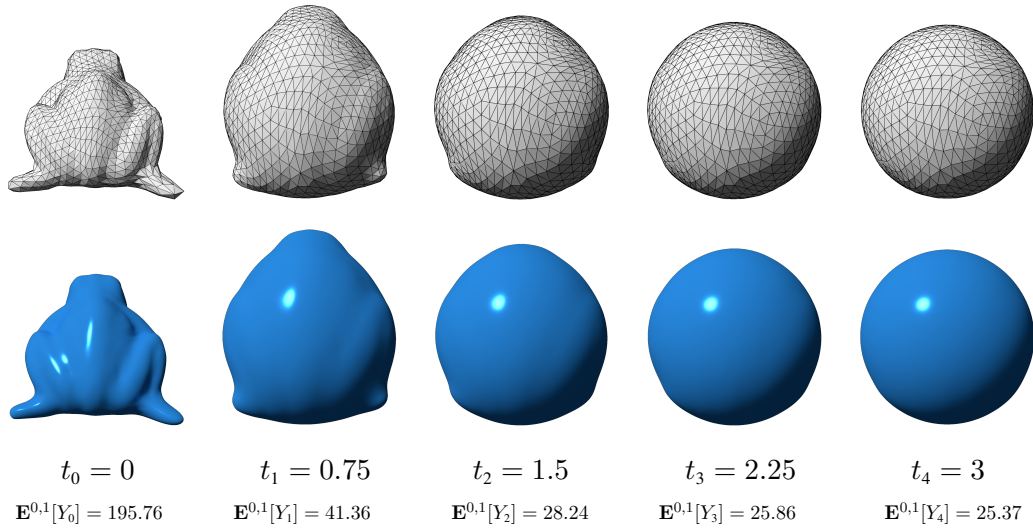


Figure 4.14.: Evolution of a frog under  $H^2_\Delta$ -Willmore flow towards a sphere with  $|\mathcal{I}_v| = 1028$ ,  $h = 0.112182$  and  $\tau = 0.75$ .

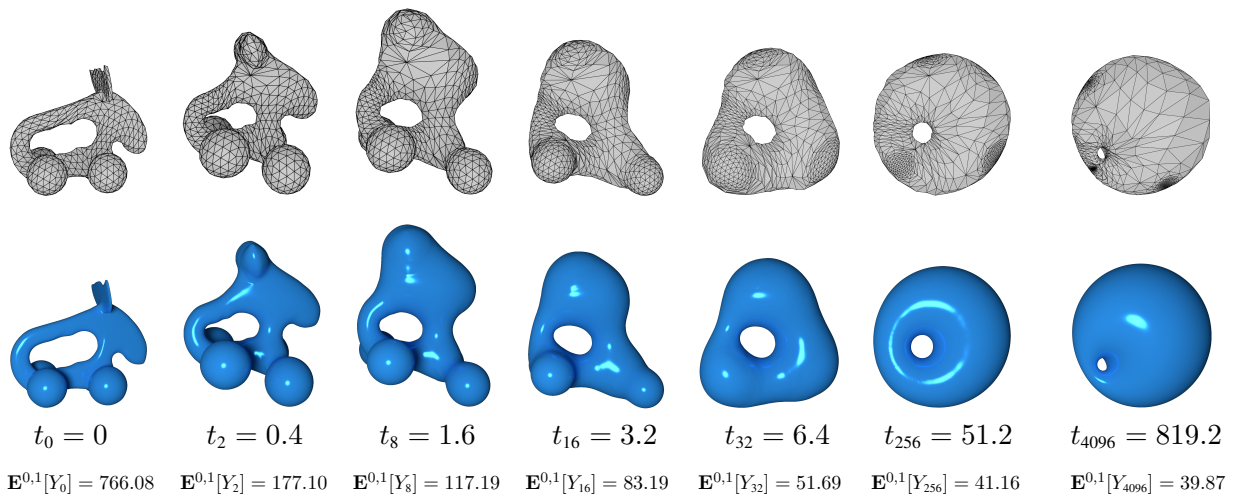


Figure 4.15.: Evolution of an elk toy under  $H^2_\Delta$ -Willmore flow towards a Möbius transformed Clifford torus with  $|\mathcal{I}_v| = 900$ ,  $h = 0.22$  and  $\tau = 0.2$ .





## 5. Geodesics, Bézier Curves and Splines in Subdivision Shell Space

In this chapter we apply the concept of *time-discrete geodesic calculus* to the *space of subdivision shells*. Furthermore, we extend the concept of *Bézier curves* to the corresponding subdivision shell space by generalizing the *de Casteljau algorithm* and define *B-splines* and *cardinal splines* in shape space by gluing together piecewise Bézier curves in a smooth way.

Curves and surfaces in Euclidean space consist of points in  $R^d$  ( $d \geq 2$ ) and the natural Euclidean metric is effective in measuring the distance between two points. In contrast, shape spaces consist of curves, surfaces, images or volumetric objects and every shape has to be understood as a point in space. Hence, the Euclidean distance is not appropriate anymore to measure the distance between two objects. In computer graphics, a common approach is to rewrite the problem via local geometric quantities such as edge lengths and dihedral angles of triangulated surfaces [139, 64]. In this setup, interpolating and morphing shapes is done by linear interpolation of edge lengths and dihedral angles and an optimal fitting step. Another approach, which incorporates more information of the underlying shape space, considers the problem as a gradient flow of a corresponding energy that measures the mismatch between a template and a reference shape [29, 53].

The dominating approach in the literature treats the space of shapes as a *Riemannian manifold*. Here, the distance between two objects in shape space is measured with respect to the underlying *Riemannian metric* and a path connecting two objects is the minimizer of all possible paths called a *geodesic*. Solving the corresponding geodesic equations involves a time stepping method. On the other hand, geodesic paths (i.e. the path connecting two shapes with minimal distance) can also be approximated via the minimization of discretized path length [127] or path energy [65]. Here, the main challenge consists in designing respectively approximating a suitable Riemannian metric which induces a notion of distance. We follow the approach of *time-discrete geodesic calculus in shell space* developed in [78, 77, 79]. This approach relies on a more general *natural time discretization* by Rumpf, Wirth and co-authors developed in the last decade [119, 118, 140, 141, 120, 142, 121, 77, 122, 123, 79], which constructs a corresponding energy that approximates the squared Riemannian distance and computes geodesic paths by minimizing the path energy. The concept of natural time discretization is linked to variational integrators [75] used in mechanics [95] and can be seen as a space-time discretization (see e.g. [69]). From a mathematical perspective, the physically motivated continuous energy of [78, 77, 79] relies on analytical results for smooth surfaces in three-dimensional nonlinear elasticity theory which are obtained by the so-called  $\Gamma$ -limit [62, 63, 61, 93].

For other approaches to Riemannian metrics for surfaces see [88, 96, 91, 92, 13, 84, 12, 23, 2] respectively for a Riemannian approach to curves see [102, 129, 133, 130, 134] and for volumetric objects, such as images, see [65, 60, 48, 17, 14]. The underlying shape spaces have again an own structure, e.g. curvature of shape space. For an exploration of the structure of space shapes see e.g. [96, 79]. Furthermore, in [79] and [143] the eigenvectors of the Hessian of the elastic energy were used, following [80, 136], to perform so-called animation without animating, respectively to perform a principle component analysis in the space of shells.

*Bézier curves* as well as *B-splines* and *cardinal splines* are a fundamental tool in *applied geometry*. Applications range from vector graphics to CAD, from computer graphics or animation to computational engineering. All three concepts are linked by the *de Casteljau algorithm* [58] which is a robust and efficient approach for drawing Bézier curves. In Euclidean space, the de Casteljau algorithm constructs polynomials by a weighted recursive combination of straight lines. In the context of Riemannian calculus, where straight lines correspond to geodesics, Bézier curves have already been investigated by generalizing the de Casteljau algorithm, see e.g. [113, 67, 54, 1]. The evaluation at any parameter position of Bézier curves always involves all input shapes. Therefore, one usually considers B-splines which solve the problem that the evaluation depends on all input shapes. Bézier curves and B-splines do not interpolate all input shapes which is a desired property in applications. Cardinal splines interpolate all input shapes and have only local dependence of input shapes but they are less smooth than equivalent B-splines. Here, we generalize B-splines and Cardinal splines to shape spaces.

Another approach is to generalize the *natural splines* from Euclidean space to shape space, see e.g. [105, 39, 24, 66, 135, 114]. A straightforward approach is to generalize the gluing of cubic Bézier segments to obtain the smoothness of cubic B-splines and the interpolation property of cardinal splines but the evaluation depends on all input shapes. In contrast to *B-splines* and *cardinal splines*, this results in a global system of constraints which is very hard to solve in arbitrary shape spaces (in Euclidean space efficient algorithms are known [58]). Therefore, the main approach in the literature is to generalize the following observation: natural splines are the minimizers of the energy  $\int_0^1 |\ddot{y}(t)|^2 dt$  in Euclidean space among all curves  $(y(t))_{t \in [0,1]}$  with prescribed supporting points  $y(t_j) = y^j$  at times  $t_j \in [0, 1]$  for  $j = 0, \dots, J$  and appropriate boundary conditions. This results in a global system involving all input shapes and all desired parameter positions. For an approach to construct higher order Riemannian polynomials in the context of polynomial regression on Riemannian manifolds see [81].

A crucial step for the numerical treatment of the space of shells or surfaces is a suitable spatial discretization of shells or surfaces. In [78, 77, 79] the continuous elasticity energy functional is discretized by applying concepts from discrete differential geometry [72, 71] and discrete exterior calculus [43, 44]. A discrete shell model for triangular meshes, which consists of a membrane and a bending term, is derived by transferring the concepts of first and second fundamental forms to triangular meshes instead of discretizing these by a finite element approach. Here, the critical part is the bending term which requires  $C^1$ -elements. A non-conforming discretization for the bending term is derived in [16] with the Crouzeix-Raviart-element. We use subdivision surfaces as discretization tool such that our approach combines higher order (time) paths in shape space (Bézier curves and splines) with higher order  $C^1$ -elements.

In Section 5.1 we review the time-discrete geodesic calculus developed in [123]. Then, we introduce generalized Bézier curves via the de Casteljau algorithm and define B-splines and cardinal splines in shape space in Section 5.2. The introduced shell model of [78, 77, 79] is described in Section 5.3 and discretized with subdivision surfaces. The detailed implementation and further algorithms are discussed in Section 5.4 and we present results in Section 5.5.

## 5.1. Time-Discrete Geodesics Calculus in Shape Space

In this section we review the *variational time discretization of geodesic calculus* developed in [123] by Rumpf and Wirth, which includes the notions of *discrete geodesics*, *discrete logarithm* and *exponential map*, and *discrete parallel transport*. Let us remark that we consider infinite-dimensional shape spaces in the continuous setting and existence of geodesics for complete Riemannian manifolds is not

guaranteed by the Hopf-Rinow theorem (see [78]). A detailed discussion of existence is beyond the scope of this chapter and we refer to [123]. In general, for infinite-dimensional shape spaces a corresponding analysis has to be performed. In Section 5.3 we will introduce the corresponding discrete setting based on the isogeometric subdivision method. Here, the existence is guaranteed following the results in [123]. In the following we will define the tools of time-continuous and time-discrete geodesic calculus and assume the existence.

**Time-continuous geodesic calculus** We denote by  $\mathcal{S}$  a *shape space* which can have various form, for example it can consist of images [17, 54], two-dimensional surfaces embedded in  $\mathbb{R}^3$  [77, 79]. Furthermore, for a shape  $y \in \mathcal{S}$  we denote the corresponding *tangent space* by  $\mathcal{T}_y\mathcal{S}$ . Let  $(\mathcal{S}, \mathcal{G})$  be a smooth, complete *Riemannian manifold* with *Riemannian metric*  $\mathcal{G} : \mathcal{T}_y\mathcal{S} \times \mathcal{T}_y\mathcal{S} \rightarrow \mathbb{R}$  for  $y \in \mathcal{S}$ . Given a smooth path  $(y(t))_{t \in [0,1]}$ , the *path length* is defined by

$$\mathcal{L}[(y(t))_{t \in [0,1]}] = \int_0^1 \sqrt{\mathcal{G}_{y(t)}(\dot{y}(t), \dot{y}(t))} dt, \quad (5.1)$$

where  $\dot{y}(t) \in \mathcal{T}_y\mathcal{S}$  denotes the derivative with respect to time  $t$ . Given two shapes  $y_A$  and  $y_B$  in  $\mathcal{S}$ , the minimizing path  $(y(t))_{t \in [0,1]}$  of (5.1) is a *geodesic* and

$$\text{dist}(y_A, y_B) = \mathcal{L}[(y(t))_{t \in [0,1]}] = \min_{\substack{(\tilde{y}(t))_{t \in [0,1]} \subset \mathcal{S} \\ \tilde{y}(0)=y_A, \tilde{y}(1)=y_B}} \mathcal{L}[(\tilde{y}(t))_{t \in [0,1]}]$$

denotes the corresponding *Riemannian distance*, i.e. the distance is the minimal path length. In general the existence of geodesics, and hence the overall concept, is not guaranteed and a detailed study has to be performed. Then, we define the *path energy* of a path  $(y(t))_{t \in [0,1]} \subset \mathcal{S}$  by

$$\mathcal{E}[(y(t))_{t \in [0,1]}] = \int_0^1 \mathcal{G}_{y(t)}(\dot{y}(t), \dot{y}(t)) dt. \quad (5.2)$$

A minimizer of (5.1) is also a minimizer of (5.2) where the constant speed property  $\mathcal{G}_{y(t)}(\dot{y}(t), \dot{y}(t)) = \mathcal{L}^2[(y(t))_{t \in [0,1]}]$  holds in these cases (see [123]). This is due to the fact that the covariant derivative  $\nabla_{\dot{y}(t)}\dot{y}(t) = 0$  for all  $t \in [0, 1]$  such that  $\frac{d}{dt}\mathcal{G}_{y(t)}(\dot{y}(t), \dot{y}(t)) = \mathcal{G}_{y(t)}(\nabla_{\dot{y}(t)}\dot{y}(t), \dot{y}(t)) = 0$  respectively  $\mathcal{G}_{y(t)}(\dot{y}(t), \dot{y}(t)) = c^2$  for fixed  $c \in \mathbb{R}$ . Again, for given shapes  $y_A$  and  $y_B$  in  $\mathcal{S}$  we define the *geodesic interpolation* at  $t \in [0, 1]$  by

$$\mathcal{I}(y_A, y_B, t) = y(t) = \left( \begin{array}{c} \arg \min \\ (\tilde{y}(s))_{s \in [0,1]} \subset \mathcal{S}, \\ \tilde{y}(0)=y_A, \tilde{y}(1)=y_B \end{array} \mathcal{E}[(\tilde{y}(s))_{s \in [0,1]}] \right) (t).$$

Here, we used the path energy for the definition of geodesic interpolation because in the discrete setting the corresponding discrete geodesic will be defined via the discrete path energy. The *logarithm map*

$$\mathcal{LOG}_{y_A}[y_B] = v,$$

assigns a tangent vector  $v \in \mathcal{T}_{y_A}\mathcal{S}$  to a shape  $y_B$  with respect to a shape  $y_A$ , where  $v = \dot{y}(0)$  is the initial velocity of a geodesic  $(y(t))_{t \in [0,1]}$  connecting  $y_A$  with  $y_B$ . The inverse operation is known as

## 5. Geodesics, Bézier Curves and Splines in Subdivision Shell Space

---

the *exponential map*, i.e. a shape  $y_A$  and a tangent vector  $v \in \mathcal{T}_{y_A}\mathcal{S}$  are mapped onto the end shape  $y_B$  of a geodesic starting at  $y_A$  with initial speed  $v$  denoted by

$$\mathcal{E}\mathcal{X}\mathcal{P}_{y_A}[v] = y_B,$$

where  $v = \dot{y}(0)$  is again the initial velocity of this geodesic  $(y(t))_{t \in [0,1]}$  connecting  $y_A$  with  $y_B$ . Finally, the *parallel transport* of a given vector  $v_0$  along a path  $(y(t))_{t \in [0,1]}$  (not necessarily geodesic) is the resulting vector  $v(1) = v_1$  from the solution of  $\nabla_{\dot{y}(t)}v(t) = 0$  for  $t \in [0, 1]$  and initial data  $v(0) = v_0$  denoted by

$$\mathcal{P}_{(y(t))_{t \in [0,1]}}[v_0] = v_1,$$

where  $\nabla_{\dot{y}}$  denotes the covariant derivative via the Levi-Civita connection  $\nabla$  (see e.g. [78]).

**Time-discrete geodesic calculus** We will now introduce the discrete counterparts to the previous time-continuous tools in shape space. Let us now sample the continuous path  $(y(t))_{t \in [0,1]}$  at time steps  $t_k = k\tau$  for  $k = 0, \dots, K$  and  $\tau = \frac{1}{K}$ , defining  $y_k = y(t_k) \in \mathcal{S}$  where  $K \geq 2$ . In this case we obtain the estimates

$$\mathcal{L}[(y(t))_{t \in [0,1]}] \geq \sum_{k=1}^K \text{dist}(y_{k-1}, y_k)$$

and

$$\mathcal{E}[(y(t))_{t \in [0,1]}] \geq \frac{1}{\tau} \sum_{k=1}^K \text{dist}^2(y_{k-1}, y_k),$$

where equality holds for geodesic paths due to the constant speed property. The first estimate is straightforward, and application of Jensen's inequality yields the second estimate

$$\sum_{k=1}^K \text{dist}^2(y_{k-1}, y_k) \leq \sum_{k=1}^K \tau \int_{(k-1)\tau}^{k\tau} \mathcal{G}_{y(t)}(\dot{y}(t), \dot{y}(t)) dt \leq \tau \mathcal{E}[(y(t))_{t \in [0,1]}].$$

The starting point of [123] is now the local approximation of the squared Riemannian distance  $\text{dist}^2$  by a smooth functional  $\mathcal{W} : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$ . Here the key act is that  $\mathcal{W}$  has to fulfill

$$\text{dist}^2(y, \tilde{y}) = \mathcal{W}[y, \tilde{y}] + \text{dist}^3(y, \tilde{y}), \quad (5.3)$$

where  $\mathcal{W}$  has to fulfill additional properties (again see [123] for details). In general,  $\mathcal{G}_y = \frac{1}{2}\mathcal{W}_{,22}[y, y]$  implies the previous statement for smooth  $\mathcal{G}$  and  $\mathcal{W}$ . Note that  $\mathcal{W}$  is not required to be symmetric. Now, we have everything at hand to introduce the *discrete path length*

$$\mathbf{L}[y_0, \dots, y_K] = \sum_{k=1}^K \sqrt{\mathcal{W}[y_{k-1}, y_k]} \quad (5.4)$$



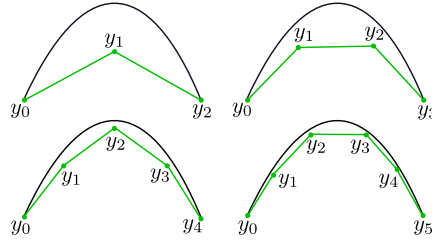


Figure 5.1.: Conceptual sketch of time-discrete geodesics for  $K = 2, 3, 4, 5$  where the black line is the time-continuous geodesic  $(y(t))_{t \in [0,1]}$ .

and the *discrete path energy*

$$\mathbf{E}[y_0, \dots, y_K] = K \sum_{k=1}^K \mathcal{W}[y_{k-1}, y_k] \quad (5.5)$$

of a *discrete path*  $(y_0, \dots, y_K)$ . For given input shapes  $y_A$  and  $y_B$  in  $\mathcal{S}$  we call the discrete path  $(y_0, \dots, y_K)$  a *discrete geodesic*, if  $(y_0, \dots, y_K)$  is a minimizer of the discrete path energy (5.5), i.e.  $(y_0, \dots, y_K)$  fulfills the necessary condition

$$K \sum_{k=1}^{K-1} (\mathcal{W}_{,2}[y_{k-1}, y_k] + \mathcal{W}_{,1}[y_k, y_{k+1}]) = 0$$

with  $y_0 = y_A$  and  $y_K = y_B$  (see Figure 5.1). Here,  $\mathcal{W}_{,1}[y, \tilde{y}]$  denotes the variation to a direction  $\Phi$  with respect to the first argument  $\mathcal{W}_{,1}[y, \tilde{y}] = \left(\frac{d}{ds} \mathcal{W}[y + s\Phi, \tilde{y}]\right)_{s=0}$  and analogously  $\mathcal{W}_{,2}[y, \tilde{y}]$  denotes the variation with respect to the second argument. The existence of discrete geodesics and thus the well-posedness of the concept has been shown in Theorem 4.3 in [123] and convergence of discrete geodesics to continuous geodesics for  $K \rightarrow \infty$  has been shown in Theorems 4.8-4.10 in [123]. Note that if  $(y_0, \dots, y_K)$  is a discrete geodesic, then so is any subsegment  $(y_i, y_{i+1}, \dots, y_{i+k})$ . Let us remark that in the discrete setting there is a difference in minimizing (5.4) and (5.5). Minimizers of the discrete path length (5.4) are in general not related to discrete geodesics (and thus also not to continuous geodesics as  $K \rightarrow \infty$ ) as discussed in Figure 2.3 in [122]. The reason is that minimizers of (5.4) can take short-cuts (e.g. for end-points close to each other on a curved two-dimensional manifold embedded in  $\mathbb{R}^3$  but separated by the ambient space, the minimizer of (5.4) will take a short-cut through the ambient space by putting discrete points close to the end points) which is prohibited for minimizers of (5.5) due to the equidistribution of points along discrete geodesics (see Theorem 4.8 in [123]). Then a *discrete geodesic interpolation* for  $0 < k < K$  is defined by

$$\mathbf{I}^K(y_A, y_B, k) = y_k = \left( \arg \min_{\substack{y_0, \dots, y_K \in \mathcal{S}, \\ y_0 = y_A, y_K = y_B}} \mathbf{E}[y_0, \dots, y_K] \right)_k.$$

Given a continuous geodesic  $(y(t))_{t \in [0,1]}$  with  $y(0) = y_A$  and  $y(1) = y_B$  and a discrete geodesic  $(y_0, \dots, y_K)$  with  $y_0 = y_A$  and  $y_K = y_B$ , we may view  $y_1 - y_0$  as the discrete counterpart to  $\tau \dot{y}(0)$  for  $\tau = \frac{1}{K}$ . Motivated by the fact that  $\left(\frac{1}{K}\right) \mathcal{LOG}_{y_A}[y_B] = \tau \dot{y}(0)$  we hence give the following definition

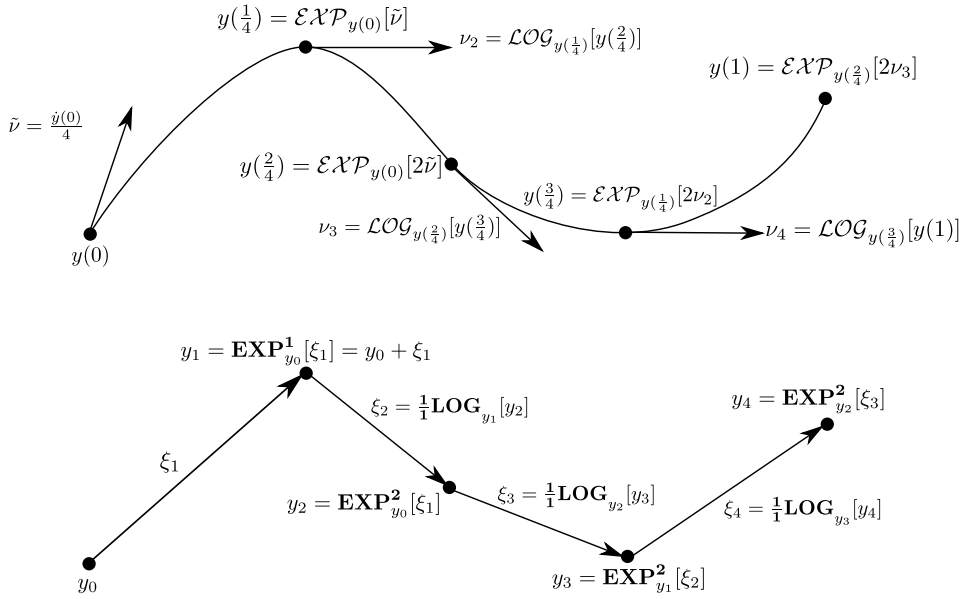


Figure 5.2.: Conceptual difference between continuous (top) and discrete (bottom) logarithm and exponential map. In contrast to the continuous setting, where  $\nu_k \in \mathcal{T}_{y(\frac{k}{K})}\mathcal{S}$  are tangent vectors, in the discrete setting  $\xi_k = y_k - y_{k-1}$  are displacements.

of a discrete logarithm. Suppose we have a discrete geodesic unique  $(y_0, \dots, y_K)$  with  $y_0 = y_A$  and  $y_K = y_B$ , we define the *discrete logarithm* by

$$\frac{1}{K}\mathbf{LOG}_{y_A}[y_B] = y_1 - y_0 = \xi,$$

where we call  $\xi$  a *displacement* which is the discrete counterpart to the tangent vector. As in the continuous case, the discrete logarithm can be considered as the linear representation of the nonlinear variation  $y_B$  to  $y_A$ . The convergence of the discrete logarithm to the continuous logarithm, for instance for a sequence of successively refined discrete geodesics  $\frac{1}{K}\mathbf{LOG}_{y_A}[y_B] \rightarrow \mathcal{L}\mathcal{O}\mathcal{G}_{y_A}[y_B]$  for  $K \rightarrow \infty$ , was proven under additional assumptions in Theorem 5.1 in [123]. As already explained, in the continuous case the exponential maps a shape  $y$  and a tangential vector  $\nu \in \mathcal{T}_y\mathcal{S}$  to a shape  $\tilde{y}$ , where a geodesic connects  $y$  and  $\tilde{y}$  with initial velocity  $\nu$ . The definition of the discrete exponential will result from our definition of the discrete path energy. Consider a discrete geodesic of length  $K = 2$ , i.e.  $(y_0, y_1, y_2)$  fulfills the condition

$$0 = 2\mathcal{W}_{2,2}[y_0, y_1] + 2\mathcal{W}_{1,1}[y_1, y_2].$$

Let us assume for a moment that  $y_0 = y$ , the shape  $y_2$  depends on  $y_1$ , and that  $y_1 = y + \xi \in \mathcal{S}$  for any sufficiently small displacement  $\xi$ , then  $y_2 = y_2[y, \xi]$  and

$$0 = 2\mathcal{W}_{2,2}[y, y + \xi] + 2\mathcal{W}_{1,1}[y + \xi, y_2[y, \xi]], \quad (5.6)$$

i.e. for given  $y \in \mathcal{S}$  and  $\xi \in \mathcal{T}_y\mathcal{S}$  sufficiently small such that  $y + \xi \in \mathcal{S}$ , the solution  $y_2[y, \xi] \in \mathcal{S}$  of (5.6) gives us a discrete counterpart to  $\mathcal{E}\mathcal{X}\mathcal{P}_y[2\xi] = y_2$ . Obviously,  $\mathcal{E}\mathcal{X}\mathcal{P}_{y_A}[\frac{k}{K}\nu] = y(\frac{k}{K})$  holds for

$k = 0, \dots, K$  if  $(y(t))_{t \in [0,1]}$  is a geodesic with  $y(0) = y_A$ ,  $y(1) = y_B$  and  $\nu = \dot{y}(0) \in \mathcal{T}_{y_A} \mathcal{S}$ . We now aim at approximating  $\mathcal{E}\mathcal{X}\mathcal{P}_{y_A}[k \cdot]$  via a discrete counterpart  $\mathbf{EXP}_{y_A}^k$ . Therefore we consider the recursive relations

$$\begin{aligned} y\left(\frac{1}{K}\right) &= \mathcal{E}\mathcal{X}\mathcal{P}_y[\tilde{\nu}] = \left(\frac{1}{1} \mathcal{L}\mathcal{O}\mathcal{G}_y\right)^{-1}[\tilde{\nu}] \\ y\left(\frac{2}{K}\right) &= \mathcal{E}\mathcal{X}\mathcal{P}_y[2\tilde{\nu}] = \left(\frac{1}{2} \mathcal{L}\mathcal{O}\mathcal{G}_y\right)^{-1}[\tilde{\nu}] \\ y\left(\frac{k}{K}\right) &= \mathcal{E}\mathcal{X}\mathcal{P}_y[k\tilde{\nu}] = \mathcal{E}\mathcal{X}\mathcal{P}_{y\left(\frac{k-2}{K}\right)}[2\nu_{k-1}] \\ &\text{where } \nu_{k-1} = \mathcal{L}\mathcal{O}\mathcal{G}_{y\left(\frac{k-2}{K}\right)}\left[y\left(\frac{k-1}{K}\right)\right] \end{aligned}$$

for given  $y \in \mathcal{S}$ ,  $\nu \in \mathcal{T}_y \mathcal{S}$  where  $\tilde{\nu} = \frac{\nu}{K} \in \mathcal{T}_y \mathcal{S}$ . Then, consider a  $y_0 \in \mathcal{S}$  and a tangent vector  $\xi \in \mathcal{T}_{y_0} \mathcal{S}$  sufficiently small with  $y_0 + \xi \in \mathcal{S}$  we define the *discrete exponential map* recursively as follows

$$\begin{aligned} y_1 &= \mathbf{EXP}_{y_0}^1[\xi] = \frac{1}{1} \mathbf{LOG}_{y_0}^{-1}[\xi] = y_0 + \xi \\ y_2 &= \mathbf{EXP}_{y_0}^2[\xi] = \frac{1}{2} \mathbf{LOG}_{y_0}^{-1}[\xi] \\ y_k &= \mathbf{EXP}_{y_0}^k[\xi] = \mathbf{EXP}_{\mathbf{EXP}_{y_0}^{k-2}(\xi)}^2[\xi_{k-1}] = \mathbf{EXP}_{y_{k-2}}^2[\xi_{k-1}], \\ &\text{where } \xi_{k-1} = \frac{1}{1} \mathbf{LOG}_{y_{k-2}}[y_{k-1}] = y_{k-1} - y_{k-2} \end{aligned}$$

and

$$\mathbf{EXP}_{y_{k-2}}^2[\xi_{k-1}] = \arg \min_{\tilde{y} \in \mathcal{S}} (\mathcal{W}_{,2}[y_{k-2}, y_{k-2} + \xi_{k-1}] + 2\mathcal{W}_{,1}[y_{k-2} + \xi_{k-1}, \tilde{y}]). \quad (5.7)$$

Convergence to the continuous exponential map has been proven in Theorem 5.10 in [123]. For the conceptual difference of continuous and discrete logarithm and exponential map we refer to Figure 5.2.

Finally, we want to define the discrete counterpart of the parallel transport. *Schild's ladder* ([55, 87]) is a well-known first-order approximation of parallel transport already in the continuous case, which is based on the construction of a sequence of geodesic parallelograms (see Figure 5.3). Given a path  $(y(t))_{t \in [0,1]}$  and a tangent vector  $\nu_{k-1} \in \mathcal{T}_{y((k-1)\tau)} \mathcal{S}$  the approximation of the parallel transported vector at time  $k\tau$  via a geodesic parallelogram can be expressed as

$$\begin{aligned} y_{k-1}^p &= \mathcal{E}\mathcal{X}\mathcal{P}_{y((k-1)\tau)}[\nu_{k-1}] \\ y_k^c &= \mathcal{E}\mathcal{X}\mathcal{P}_{y_{k-1}^p}[\mathcal{L}\mathcal{O}\mathcal{G}_{y_{k-1}^p}[y(k\tau)]] \\ y_k^p &= \mathcal{E}\mathcal{X}\mathcal{P}_{y((k-1)\tau)}[2\mathcal{L}\mathcal{O}\mathcal{G}_{y((k-1)\tau)}[y_k^c]] \text{ where} \\ \nu_k &= \mathcal{L}\mathcal{O}\mathcal{G}_{y(k\tau)}[y_k^p]. \end{aligned}$$

Let  $(y_0, \dots, y_K)$  be a discrete path in  $\mathcal{S}$  with  $y_k - y_{k-1}$  sufficiently small for  $k = 1, \dots, K$ . Then the *discrete parallel transport* of a sufficiently small displacement  $\xi_0$  at  $y_0$  along  $(y_0, \dots, y_K)$  is defined

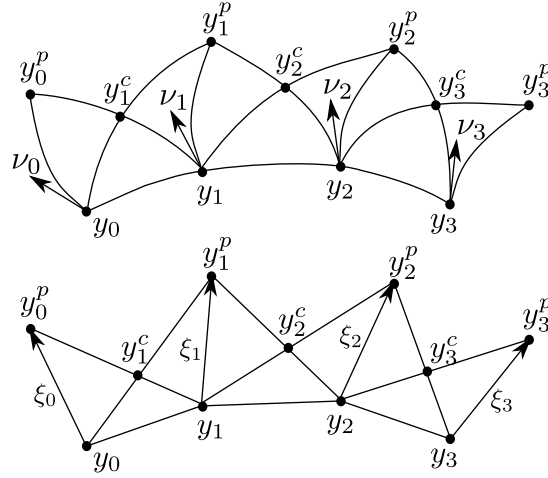


Figure 5.3.: Conceptual difference between continuous (top) and discrete (bottom) parallel transport. In contrast to continuous setting, where  $\nu_k \in \mathcal{T}_{y_k} \mathcal{S}$  are tangent vectors, in the discrete setting  $\xi_k = y_k^p - y_k$  are displacements and  $\mathbf{P}_{y_{k+1}, y_k}^K[\xi_k] = \xi_{k+1}$  maps a displacement  $\xi_k$  from a shape  $y_k$  to a displacement  $\xi_{k+1}$  at shape  $y_{k+1}$ .

for  $k = 1, \dots, K$  via the iteration

$$\begin{aligned} y_{k-1}^p &= y_{k-1} + \xi_{k-1} \\ y_k^c &= y_{k-1}^p + \frac{1}{2} \mathbf{LOG}_{y_{k-1}^p}[y_k] \\ y_k^p &= \mathbf{EXP}_{y_{k-1}^p}^2[y_k^c - y_{k-1}^p] \\ \xi_k &= y_k^p - y_k, \end{aligned}$$

where  $\xi_k$  is the transported displacement at  $y_k$ . We denote this process by

$$\mathbf{P}_{(y_K, \dots, y_0)}^K[\xi_0] = \xi_K.$$

The well-definedness of the above iteration as well as the convergence against the continuous parallel transport are treated in Theorem 5.11 in [123]. Last but not least, we compute the discrete parallel transport of a displacement  $\xi_{k-1}$  from a shape  $y_{k-1}$  to  $y_k$ , where  $\xi_{k-1}$  can be large, by first computing the intermediate shape  $y_k^c$  for  $M \geq 2$

$$y_k^c = \mathbf{I}^{2M}(y_{k-1} + \xi_{k-1}, y_k, M)$$

between  $y_{k-1}^p = y_{k-1} + \xi_{k-1}$  and  $y_k$ . In a second step we compute the discrete extrapolated shape by

$$y_k^p = \mathbf{EXP}_{\mathbf{I}^K(y_{k-1}, y_k^c, K-1)}^{M+1}[y_k^c - \mathbf{I}^K(y_{k-1}, y_k^c, M-1)]$$

and obtain the parallel transport shape displacement  $\xi_k = y_k^p - y_k$ , i.e.  $\mathbf{P}_{(y_k, y_{k-1})}^K[\xi_{k-1}] = \xi_k$  (see Figure 5.3).

## 5.2. Bézier Curves and Splines in Shape Space

Now, we consider Bézier curves in shape space [54] and extend quadratic and cubic Bézier curves to quadratic respectively cubic B-splines in shape space. Furthermore, we use the relation between cubic Bézier curves and cubic Hermite curves [58] to define cardinal splines in shape space. Let us remark, that the existence of the concepts in infinite-dimensional shapes spaces is again not guaranteed as in the previous section. But in the discrete spatial setting, introduced in Section 5.3, the existence of geodesics discussed in [123] will imply the existence of the concepts introduced in this section.

**Time-discrete Bézier curves in shape space** Consider a set of control shapes  $(y_0^0, \dots, y_n^0)$  with  $y_j^0 \in \mathcal{S}$  for  $j = 0, \dots, n$  and the mapping

$$\mathcal{B} : \underbrace{\mathcal{S} \times \dots \times \mathcal{S}}_{n+1} \times [0, 1] \rightarrow \mathcal{S}$$

which is recursively defined via the *de Casteljau algorithm*

$$\mathcal{B}(y_i, \dots, y_j, t) = \mathcal{I}(\mathcal{B}(y_i, \dots, y_{j-1}, t), \mathcal{B}(y_{i+1}, \dots, y_j, t), t)$$

for  $i, j \in \{0, \dots, n\}$  and  $i < j$ . In other words for  $(y_0^0, \dots, y_n^0)$  and fixed  $t \in [0, 1]$  we compute for  $j = 1, \dots, n$  and  $i = j, \dots, n$  the shapes

$$y_i^j = \mathcal{I}(y_{i-1}^{j-1}, y_i^{j-1}, t)$$

and after  $n$  steps we obtain

$$\mathcal{B}(y_0^0, \dots, y_n^0, t) = y_n^n.$$

We call the resulting curve

$$(\mathcal{B}(y_0^0, \dots, y_n^0, t))_{t \in [0, 1]}$$

a *Bézier curve in shape space of degree  $n$* .

Compared to the Euclidean case the shapes  $y_i^j$  do not lie on a straight line but on a geodesic  $(\mathcal{I}(y_{i-1}^{j-1}, y_i^{j-1}, t))_{t \in [0, 1]}$  connecting  $y_{i-1}^{j-1}$  with  $y_i^{j-1}$ . Again, in the Euclidean setting, it is well-known that cubic Hermite curves can be expressed with cubic Bézier curves (see [58]). Assume we have two points  $p_0, p_1 \in \mathbb{R}^d$  ( $d \geq 2$ ) with tangents  $m_0$  and  $m_1$  at  $p_0$  respectively  $p_1$ . Then, the corresponding cubic Hermite curve of  $p_0, m_0, m_1$  and  $p_1$  can be expressed by cubic Bézier control points  $(b_0, b_1, b_2, b_3)$  by

$$\begin{aligned} b_0 &= p_0, \\ b_1 &= p_0 + \frac{m_0}{3}, \\ b_2 &= p_1 - \frac{m_1}{3}, \\ b_3 &= p_1, \end{aligned}$$

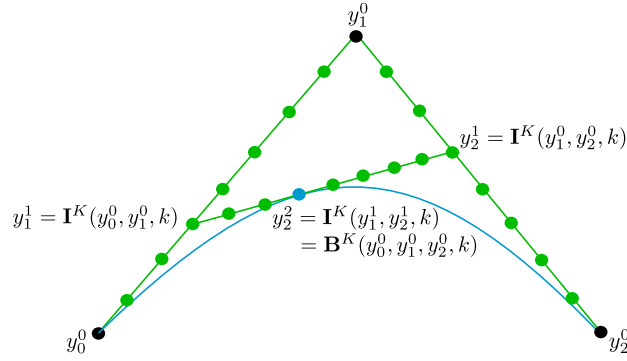


Figure 5.4.: Sketch of the discrete de Casteljau algorithm for the definition of a discrete quadratic Bézier curve in shape space.

where  $b_i \in \mathbb{R}^d$ . We will now transfer this to the shape space setting. For given shapes  $y_A, y_B \in \mathcal{S}$  and tangent vectors  $\nu_A \in \mathcal{T}_{y_A}\mathcal{S}$  and  $\nu_B \in \mathcal{T}_{y_B}\mathcal{S}$  we can compute a *cubic Hermite curve* by

$$\begin{aligned} \mathcal{H}(y_A, \nu_A, \nu_B, y_B, t) &= \mathcal{B}(y_0, y_1, y_2, y_3, t), \quad \text{where} \\ y_0 &= y_A \\ y_1 &= \mathcal{E}\mathcal{X}\mathcal{P}_{y_A} \left[ \frac{1}{3} \cdot \nu_A \right] \\ y_2 &= \mathcal{E}\mathcal{X}\mathcal{P}_{y_B} \left[ \frac{1}{3} \cdot \nu_B \right] \\ y_3 &= y_B. \end{aligned}$$

Here, we did not write a minus in front of  $\nu_B$ , because we assume that  $\nu_B$  points to the right direction in the tangent space  $\mathcal{T}_{y_B}\mathcal{S}$ .

Now, we transfer the Bézier curve to our time-discrete setup. Consider a set of control shapes  $(y_0^0, \dots, y_n^0)$  with  $y_j^0 \in \mathcal{S}$  for  $j = 0, \dots, n$  and the mapping

$$\mathbf{B}^K : \underbrace{\mathcal{S} \times \dots \times \mathcal{S}}_{n+1} \times \{0, \dots, K\} \rightarrow \mathcal{S}$$

which is recursively defined via the *discrete de Casteljau algorithm*

$$\mathbf{B}^K(y_i, \dots, y_j, k) = \mathbf{I}^K(\mathbf{B}^K(y_i, \dots, y_{j-1}, k), \mathbf{B}^K(y_{i+1}, \dots, y_j, k), k)$$

for  $i, j \in \{0, \dots, n\}$  and  $i < j$ . In other words for  $(y_0^0, \dots, y_n^0)$  and fixed  $k \in \{0, \dots, K\}$  we compute for  $j = 1, \dots, n$  and  $i = j, \dots, n$  the shapes

$$y_i^j = \mathbf{I}^K(y_{i-1}^{j-1}, y_i^{j-1}, k)$$

and after  $n$  steps we obtain

$$\mathbf{B}^K(y_0^0, \dots, y_n^0, k) = y_n^n.$$

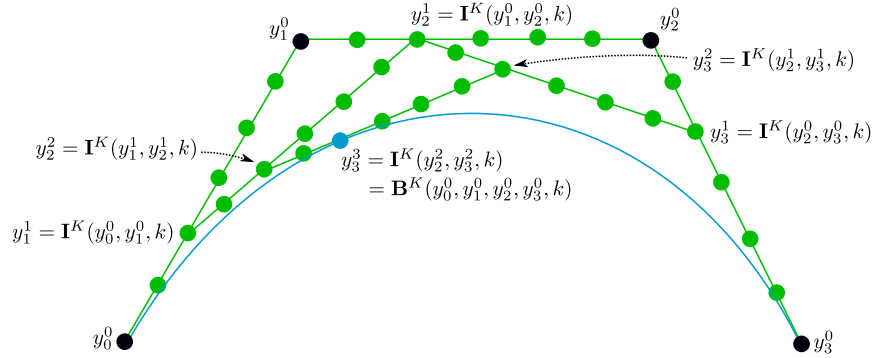


Figure 5.5.: Sketch of the discrete de Casteljau algorithm for the definition of a discrete cubic Bézier curve in shape space.

We call the resulting discrete path

$$(\mathbf{B}^K(y_0, \dots, y_n, k))_{k \in \{0, \dots, K\}}$$

a discrete Bézier curve in shape space of degree  $n$ .

In the following we will focus on the cases  $n = 2$  and  $n = 3$  known as quadratic respectively cubic Bézier curve. For  $n = 2$  (i.e. for given shapes  $(y_0^0, y_1^0, y_2^0)$ ) and given  $K > 1$ , the evaluation at  $k \in \{0, \dots, K\}$  requires the following computation (see Figure 5.4)

$$\begin{aligned} \mathbf{B}^K(y_0^0, y_1^0, y_2^0, k) &= y_2^2 \\ &= \mathbf{I}^K(y_1^1, y_2^1, k) \\ &= \mathbf{I}^K(\mathbf{I}^K(y_0^0, y_1^0, k), \mathbf{I}^K(y_1^0, y_2^0, k), k). \end{aligned}$$

From an implementation point of view we will perform a two-step algorithm. First, we compute the discrete geodesics  $y_1^1(k) = (\mathbf{I}^K(y_0^0, y_1^0, k))_{k \in \{0, \dots, K\}}$  and  $(y_2^1(k) = \mathbf{I}^K(y_1^0, y_2^0, k))_{k \in \{0, \dots, K\}}$  and then we will compute  $K$ -times the discrete geodesic  $(\mathbf{I}^K(y_1^1(k), y_2^1(k), j))_{j \in \{0, \dots, K\}}$ . The computational effort increases for  $n = 3$ . Consider the input shapes  $(y_0^0, y_1^0, y_2^0, y_3^0)$  and  $K > 1$ , then the evaluation at  $k \in \{0, \dots, K\}$  requires the following computation (see Figure 5.5)

$$\begin{aligned} \mathbf{B}^K(y_0^0, y_1^0, y_2^0, y_3^0, k) &= y_3^3 \\ &= \mathbf{I}^K(y_2^2, y_3^2, k) \\ &= \mathbf{I}^K(\mathbf{I}^K(y_1^1, y_2^1, k), \mathbf{I}^K(y_2^1, y_3^1, k), k) \\ &= \mathbf{I}^K(\mathbf{I}^K(\mathbf{I}^K(y_0^0, y_1^0, k), \mathbf{I}^K(y_1^0, y_2^0, k), k), \mathbf{I}^K(\mathbf{I}^K(y_1^0, y_2^0, k), \mathbf{I}^K(y_2^0, y_3^0, k), k), k). \end{aligned}$$

Again, from an implementation point of view we will perform a three-step algorithm (again see Figure 5.5). In general the evaluation of a Bézier curve of degree  $n$  with input shapes  $(y_0^0, \dots, y_n^0)$ , internal geodesics of order  $K$  and parameter position  $k \in \{0, \dots, K\}$  reads (i.e. the discrete de Casteljau algorithm):

```

for  $j = 1$  to  $n$  do
    for  $i = j$  to  $n$  do
    
```

```

     $y_i^j = \mathbf{I}^K[y_{i-1}^{j-1}, y_i^{j-1}, k]$ 
  end for
end for
 $\mathbf{B}^K[y_0^0, \dots, y_n^0, k] = y_n^n$ 

```

Finally, we define the *discrete cubic Hermite curve*  $(\mathbf{H}^K(y_A, \xi_A, \xi_B, y_B, k))_{k \in \{0, \dots, K\}}$  for given shapes  $y_A, y_B \in \mathcal{S}$ , displacements  $\xi_A, \xi_B$  and  $K > 1$  by

$$\mathbf{H}^K(y_A, \xi_A, \xi_B, y_B, k) = \mathbf{B}^K(y_0, y_1, y_2, y_3, k) \quad \text{where}$$

$$y_0 = y_A,$$

$$y_1 = \mathbf{EXP}_{y_A}^K \left[ \frac{1}{3K} \xi_A \right],$$

$$y_2 = \mathbf{EXP}_{y_B}^K \left[ \frac{1}{3K} \xi_B \right],$$

$$y_3 = y_B.$$

One of the main drawbacks of Bézier curves is that the evaluation at any parameter position (except  $k = 0$  and  $k = K$ ) requires all input shapes which results in a really expensive tool for high  $n$ . Therefore we will now introduce quadratic and cubic B-splines in shape space.

**Time-discrete B-splines in shape space** Consider a set of control shapes  $(y_0, \dots, y_m)$  with  $y_j \in \mathcal{S}$  for  $j = 0, \dots, m$ ,  $m > 2$  and the mapping

$$\mathcal{QBS} : \underbrace{\mathcal{S} \times \dots \times \mathcal{S}}_{m+1} \times [0, m-1] \rightarrow \mathcal{S}$$

which is defined piecewise for  $t \in [l, l+1]$  where  $l = 0, \dots, m-2$  by

$$\mathcal{QBS}(y_0, \dots, y_m, t) = \mathcal{B}(d_{2l}, d_{2l+1}, d_{2l+2}, t_l)$$

with

$$d_0 = y_0,$$

$$d_{2j+1} = y_{j+1}, \quad j = 0, \dots, m-2,$$

$$d_{2j} = \mathcal{I}(y_j, y_{j+1}, 0.5), \quad j = 1, \dots, m-2,$$

$$d_{2(m-1)} = y_m$$

and  $t_l = t - l \in [0, 1]$ . We call the resulting curve

$$(\mathcal{QBS}(y_0, \dots, y_m, t))_{t \in [0, m-1]}$$

the *quadratic B-splines in shape space*. This definition results in a curve that interpolates the first and last control shape as for Bézier curves but the evaluation process (except the computation of intermediate Bézier control shapes) requires only the computation of quadratic Bézier curves. For  $m = 2$  we would obtain the quadratic Bézier curve. The transfer to the discrete setting is as follows. Consider a set of control shapes  $(y_0, \dots, y_m)$  with  $y_j \in \mathcal{S}$  for  $j = 0, \dots, m$ ,  $m > 2$  and  $K \geq 2$ ,  $M \geq 1$  the



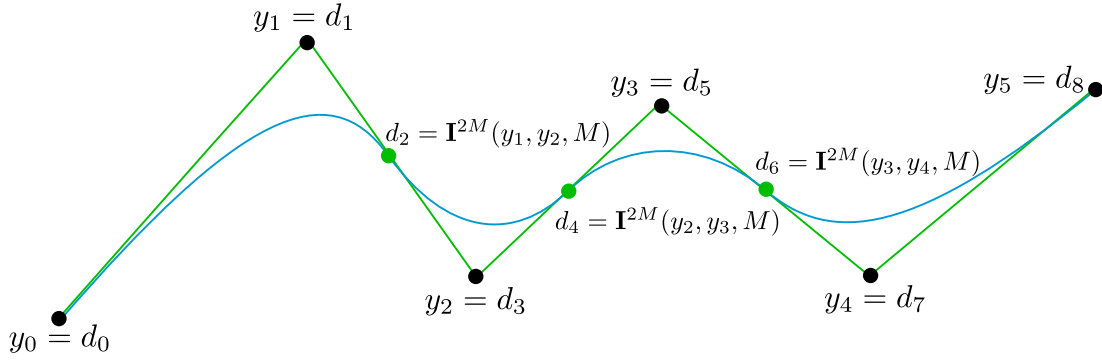


Figure 5.6.: Sketch of the construction of the Bézier control shapes for the discrete quadratic B-spline curve in shape space.

mapping

$$\mathbf{QBS}^{K,M} : \underbrace{\mathcal{S} \times \dots \times \mathcal{S}}_{m+1} \times \{0, \dots, (m-1) \cdot K\} \rightarrow \mathcal{S}$$

which is defined piecewise for  $k \in [l \cdot K, \dots, (l+1) \cdot K]$  where  $l = 0, \dots, m-2$  by

$$\mathbf{QBS}^{K,M}(y_0, \dots, y_m, k) = \mathcal{B}(d_{2l}, d_{2l+1}, d_{2l+2}, k_l)$$

with (see Figure 5.6 for an example)

$$\begin{aligned} d_0 &= y_0, \\ d_{2j+1} &= y_{j+1}, \quad j = 0, \dots, m-2, \\ d_{2j} &= \mathbf{I}^{2M}(y_j, y_{j+1}, M), \quad j = 1, \dots, m-2, \\ d_{2(m-1)} &= y_m \end{aligned}$$

and  $k_l = k - l \cdot K \in \{0, \dots, K\}$ . We call the resulting curve

$$(\mathbf{QBS}^{K,M}(y_0, \dots, y_m, k))_{k \in \{0, \dots, (m-1) \cdot K\}}$$

the *discrete quadratic B-spline in shape space*. In contrast to the Bézier curves we introduced an additional parameter  $M \geq 1$  in the definition. Here,  $M$  controls the precision of the control shape  $d_{2j}$  which lies on the geodesics between input shapes  $y_j$  and  $y_{j+1}$ .

Now, we increase the degree and consider cubic B-splines in shape space. Consider a set of control shapes  $(y_0, \dots, y_m)$  with  $y_j \in \mathcal{S}$  for  $j = 0, \dots, m$ ,  $m > 3$  and the mapping

$$\mathbf{CBS} : \underbrace{\mathcal{S} \times \dots \times \mathcal{S}}_{m+1} \times [0, m-2] \rightarrow \mathcal{S}$$

which is defined piecewise for  $t \in [l, l+1]$  and  $l = 0, \dots, m-3$  by

$$\mathbf{CBS}(y_0, \dots, y_m, t) = \mathcal{B}(d_{3l}, d_{3l+1}, d_{3l+2}, d_{3l+3}, t_l)$$

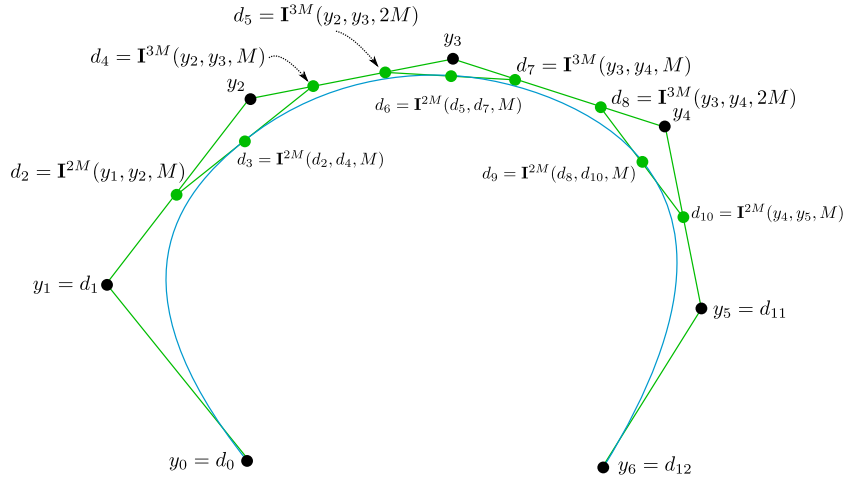


Figure 5.7.: Sketch of the construction of the Bézier control shapes for the discrete cubic B-spline curve in shape space.

with

$$\begin{aligned}
 d_0 &= y_0, \\
 d_1 &= y_1, \\
 d_2 &= \mathcal{I}(y_1, y_2, 1/2), \\
 d_{3j-1} &= \mathcal{I}(y_j, y_{j+1}, 2/3), \quad j = 2, \dots, m-3, \\
 d_{3j} &= \mathcal{I}(d_{3j-1}, d_{3j+1}, 1/2), \quad j = 1, \dots, m-3, \\
 d_{3j+1} &= \mathcal{I}(y_{j+1}, y_{j+2}, 1/3), \quad j = 1, \dots, m-4, \\
 d_{3(m-2)-2} &= \mathcal{I}(y_{m-2}, y_{m-1}, 1/2), \\
 d_{3(m-2)-1} &= y_{m-1}, \\
 d_{3(m-2)} &= y_m
 \end{aligned}$$

and  $t_l = t - l \in [0, 1]$ . We call the resulting curve

$$(\mathcal{CBS}(y_0, \dots, y_m, t))_{t \in [0, m-2]}$$

the *cubic B-spline in shape space*. Again, the curve interpolates the first and last control shape by defining  $m-2$  segments of cubic Bézier curves and gluing them together. For  $m=3$  we would obtain the cubic Bézier curve. Again, let us transfer this definition to the discrete setting. Consider a set of control shapes  $(y_0, \dots, y_m)$  with  $y_j \in \mathcal{S}$  for  $j = 0, \dots, m$ ,  $m > 3$  and  $K \geq 2$ ,  $M \geq 1$  the mapping

$$\mathcal{CBS}^{K,M} : \underbrace{\mathcal{S} \times \dots \times \mathcal{S}}_{m+1} \times \{0, \dots, (m-2) \cdot K\} \rightarrow \mathcal{S}$$

which is defined piecewise for  $k \in \{l \cdot K, \dots, (l+1) \cdot K\}$  where  $l = 0, \dots, m-3$  by

$$\mathcal{CBS}^{K,M}(y_0, \dots, y_m, k) = \mathcal{B}(d_{3l}, d_{3l+1}, d_{3l+2}, d_{3l+3}, k_l)$$

with (see Figure 5.7 for an example)

$$\begin{aligned}
 d_0 &= y_0, \\
 d_1 &= y_1, \\
 d_2 &= \mathbf{I}^{2M}(y_1, y_2, M), \\
 d_{3j-1} &= \mathbf{I}^{3M}(y_j, y_{j+1}, 2M), \quad j = 2, \dots, m-3, \\
 d_{3j} &= \mathbf{I}^{2M}(d_{3j-1}, d_{3j+1}, M), \quad j = 1, \dots, m-3, \\
 d_{3j+1} &= \mathbf{I}^{3M}(y_{j+1}, y_{j+2}, M), \quad j = 1, \dots, m-4, \\
 d_{3(m-2)-2} &= \mathbf{I}^{2M}(y_{m-2}, y_{m-1}, M), \\
 d_{3(m-2)-1} &= y_{m-1}, \\
 d_{3(m-2)} &= y_m
 \end{aligned}$$

and  $k_l = k - l \cdot K \in \{0, \dots, K\}$ . We call the resulting curve

$$(\mathbf{CBS}^{K,M}(y_0, \dots, y_m, k))_{k \in \{0, \dots, (m-2) \cdot K\}}$$

the *discrete cubic B-spline in shape space*.

Unfortunately, both curves produces visually discrete paths but do not interpolate the input (control) shapes. Therefore, we will now glue together piecewise cubic Hermite curves to obtain a curve that is less visually smooth but interpolates the input shapes.

**Time-discrete cardinal splines in shape space** Consider a set of control shapes  $(y_0, \dots, y_m)$  with  $y_j \in \mathcal{S}$  for  $j = 0, \dots, m$ ,  $m > 3$  and  $\kappa \in [0, 3]$  the mapping

$$\mathcal{CS}_\kappa : \underbrace{\mathcal{S} \times \dots \times \mathcal{S}}_{m+1} \times [0, m] \rightarrow \mathcal{S}$$

which is defined piecewise for  $t \in [k, k+1]$  where  $k = 0, \dots, m-1$  by

$$\mathcal{CS}_\kappa(y_0, \dots, y_m, t) = \mathcal{B}(d_{3k}, d_{3k+1}, d_{3k+2}, d_{3k+3}, t_k)$$

with

$$\begin{aligned}
 d_{3j} &= y_j, \quad j = 0, \dots, m, \\
 d_1 &= \mathcal{I}(y_0, y_1, \kappa/3), \\
 d_{3m-1} &= \mathcal{I}(y_{m-1}, y_m, 1 - \kappa/3) = \mathcal{I}(y_m, y_{m-1}, \kappa/3), \\
 d_{3j-1} &= \mathcal{E}\mathcal{X}\mathcal{P}_{d_{3j+1}} [2\mathcal{L}\mathcal{O}\mathcal{G}_{d_{3j+1}}[d_{3j}]], \quad j = 1, \dots, m-1, \\
 d_{3j+1} &= \mathcal{E}\mathcal{X}\mathcal{P}_{y_j}[\nu_j^p], \quad j = 1, \dots, m-1 \quad \text{where} \\
 \nu_j^p &= \mathcal{P}(\mathcal{I}(y_{j-1}, y_j, t))_{t \in [0,1]}[\nu_j^g], \\
 \nu_j^g &= \mathcal{L}\mathcal{O}\mathcal{G}_{y_{j-1}}[y_j^g], \\
 y_j^g &= \mathcal{I}(y_{j-1}, y_{j+1}, \kappa/3)
 \end{aligned}$$

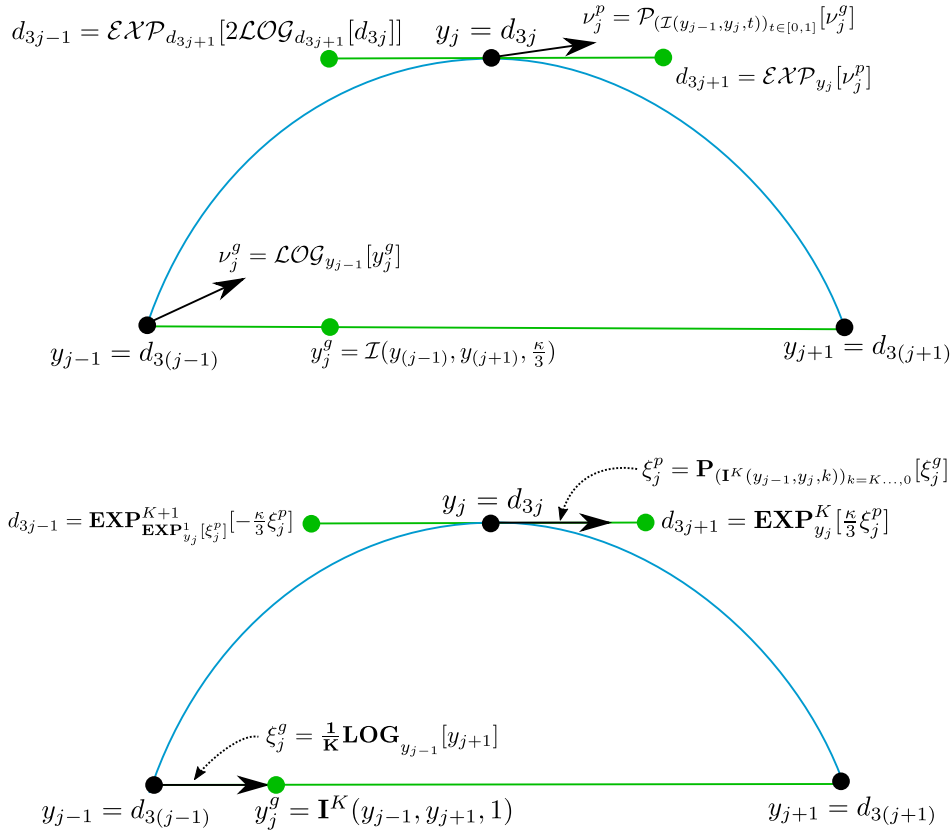


Figure 5.8.: Sketch of the construction of the control shapes  $d_{3j-1}$  and  $d_{3j+1}$  for the cardinal spline in the continuous (top) and discrete (bottom) setting.

and  $t_k = t - k \in [0, 1]$ . We call the resulting curve

$$(\mathcal{CS}_\kappa(y_0, \dots, y_m, t))_{t \in [0, m]}$$

the *cubic cardinal spline in shape space* where  $\kappa \in [0, 3]$  denotes the *tension parameter*. Because  $d_{3j} = y_j$  the curve interpolates all control shapes  $(y_0, \dots, y_m)$ .

Let us now define the corresponding discrete setting. Consider a set of control shapes  $(y_0, \dots, y_m)$  with  $y_j \in \mathcal{S}$  for  $j = 0, \dots, m$ ,  $m > 3$  and  $K \geq 2$ ,  $M \geq 1$  the mapping

$$\mathbf{CS}_\kappa^K : \underbrace{\mathcal{S} \times \dots \times \mathcal{S}}_{m+1} \times \{0, \dots, m \cdot K\} \rightarrow \mathcal{S}$$

which is defined piecewise for  $k \in \{l \cdot K, \dots, (l+1) \cdot K\}$  where  $l = 0, \dots, m-1$  by

$$\mathbf{CS}_\kappa^K(y_0, \dots, y_m, k) = \mathcal{B}(d_{3l}, d_{3l+1}, d_{3l+2}, d_{3l+3}, k_l)$$

with (see Figure 5.8 for an example)

$$d_{3j} = y_j, \quad j = 0, \dots, m,$$

$$\begin{aligned}
 d_1 &= \mathbf{EXP}_{y_0}^1 \left[ \kappa/3 \frac{1}{\mathbf{K}} \mathbf{LOG}_{y_0}[y_1] \right], \\
 d_{3m-1} &= \mathbf{EXP}_{y_m}^1 \left[ \kappa/3 \frac{1}{\mathbf{K}} \mathbf{LOG}_{y_m}[y_{m-1}] \right], \\
 d_{3j-1} &= \mathbf{EXP}_{\mathbf{EXP}_{y_j}^1[\xi_j^p]}^{\mathbf{K}+1} \left[ -\frac{\kappa}{3} \xi_j^p \right], \quad j = 1, \dots, m-1, \\
 d_{3j+1} &= \mathbf{EXP}_{y_j}^{\mathbf{K}} \left[ \frac{\kappa}{3} \xi_j^p \right], \quad j = 1, \dots, m-1, \quad \text{where} \\
 \xi_j^p &= \mathbf{P}_{(\mathbf{I}^{\mathbf{K}}(y_{j-1}, y_j, k))_{k=\{K, \dots, 0\}}}^{\mathbf{K}} \left[ \xi_j^g \right], \\
 \xi_j^g &= \frac{1}{\mathbf{K}} \mathbf{LOG}_{y_{j-1}}[y_{j+1}] = \frac{1}{1} \mathbf{LOG}_{y_{j-1}}[y_j^g], \\
 y_j^g &= \mathbf{I}^{\mathbf{K}}(y_{j-1}, y_{j+1}, 1)
 \end{aligned}$$

and  $k_l = k - l \cdot K \in \{0, \dots, K\}$ . We call the resulting curve

$$(\mathbf{CS}_{\kappa}^{\mathbf{K}}(y_0, \dots, y_m, k))_{k \in \{0, \dots, m \cdot K\}}$$

the *discrete cubic cardinal spline in shape space* with tension  $\kappa \in [0, 3]$ . For the computation of the control shapes  $d_{3j-1}$  and  $d_{3j+1}$  we refer again to the sketch in Figure 5.8. Let us remark that this is equivalent to the construction with discrete cubic Hermite splines where a segment consists of the shapes  $y_j$  and  $y_{j+1}$  and displacements  $\xi_j^p$  and  $-\xi_{j+1}^p$ .

### 5.3. Subdivision Shell Space

In the previous sections we described the conceptual framework for discrete geodesics, discrete logarithm and exponential map, discrete parallel transport, discrete Bézier curves, discrete B-splines and discrete cardinal splines. All these concepts are based on the definition of a suitable energy  $\mathcal{W} : \mathcal{S} \times \mathcal{S} \mapsto \mathbb{R}$  which approximates the squared Riemannian distance  $\text{dist}^2$ . As already stated there are many possibilities to choose such an energy functional. Here, we follow [78, 77, 79] where the authors derive an energy functional that is related to the elastic properties of thin shells by measuring membrane and bending contributions of deformations. This gives the underlying shape space the structure for shells which motivates the term shell space. Furthermore, we will discretize the smooth energy with subdivision surfaces, i.e. we define the *subdivision shell space*.

**Deformation mapping** Let  $\bar{\mathcal{M}}$  be a smooth, embedded surface in  $\mathbb{R}^3$ , then we denote by

$$\bar{\mathcal{M}}^\delta = \bar{\mathcal{M}} \times \left[ -\frac{\delta}{2}, \frac{\delta}{2} \right]$$

a *shell* with thickness  $\delta$ , i.e. a shell is a thin, curved three dimensional object where the thickness is relatively small compared the overall *mid-surface*  $\bar{\mathcal{M}}$ . Assume that there is a force effect on the boundary  $\partial \bar{\mathcal{M}}^\delta$  of  $\bar{\mathcal{M}}^\delta$ , where some part of the shell boundary is fixed, induced by a deformation  $\phi^\delta : \bar{\mathcal{M}}^\delta \rightarrow \mathbb{R}^3$  with  $\phi(\bar{\mathcal{M}}^\delta) = \mathcal{M}^\delta$ , i.e. for every point  $\bar{x} \in \bar{\mathcal{M}}^\delta$  is mapped to a point  $x \in \mathcal{M}^\delta$  by the deformation map  $\phi^\delta$ . We call  $\bar{\mathcal{M}}^\delta$  the *undeformed shell* and  $\mathcal{M}^\delta$  the *deformed shell*. The overall deformation can be measured by an elastic energy functional  $\mathcal{W}^\delta[\phi^\delta]$ . In the context of shells one usually transfers the setting to the mid-surface, i.e. instead of considering the elastic energy functional

$\mathcal{W}^\delta[\phi^\delta]$  we consider a  $\delta$ -scaled variant  $\mathcal{W}[\phi]$  where  $\phi : \bar{\mathcal{M}} \rightarrow \mathbb{R}^3$  with  $\phi(\bar{\mathcal{M}}) = \mathcal{M}$  denotes the *deformation mapping* of the mid-surface. In [78, 77, 79] Heeren and co-authors consider the following elastic energy

$$\begin{aligned} \mathcal{W}[\phi] &= \mathcal{W}[\bar{\mathcal{M}}, \phi(\bar{\mathcal{M}})] = \mathcal{W}[\bar{\mathcal{M}}, \mathcal{M}] = \delta \mathcal{W}_{\text{mem}}[\phi] + \delta^3 \mathcal{W}_{\text{bend}}[\phi] \\ &= \delta \int_{\bar{\mathcal{M}}} W_{\text{mem}}[\phi] \, da + \delta^3 \int_{\bar{\mathcal{M}}} W_{\text{bend}}[\phi] \, da \\ &= \delta \int_{\bar{\mathcal{M}}} W_{\text{mem}}[\bar{\mathcal{M}}, \mathcal{M}] \, da + \delta^3 \int_{\bar{\mathcal{M}}} W_{\text{bend}}[\bar{\mathcal{M}}, \mathcal{M}] \, da. \end{aligned}$$

The definition reflects the two predominating mechanisms by the deformation of a shell. On one hand *tangential* respectively *shear* contributions, i.e. changes of the first fundamental forms measured by the *membrane energy*  $W_{\text{mem}}$ , and on the other hand *bending* contributions are measured, i.e. changes of the second fundamental forms measured by the *bending energy*  $W_{\text{bend}}$ . Let us remark that this fits into our setting of a smooth path as follows. For a given mid-surfaces  $\bar{\mathcal{M}}$  and  $\mathcal{M}$  we define a deformation path  $(\phi(t))_{t \in [0,1]}$  where  $\phi(t, \bar{\mathcal{M}}) = \mathcal{M}(t)$ ,  $\phi(0, \bar{\mathcal{M}}) = \bar{\mathcal{M}}$  and  $\phi(1, \bar{\mathcal{M}}) = \mathcal{M}$  as the path with minimal elastic energy, which corresponds to the path energy  $\mathcal{E}$ . Here,  $(\mathcal{M}(t))_{t \in [0,1]}$  is the smooth path between  $\bar{\mathcal{M}}$  and  $\mathcal{M}$  respectively for parameterizations  $y(t)$  of  $\mathcal{M}(t)$  the smooth path is  $(y(t))_{t \in [0,1]}$  which fits perfectly into the setting described in the last sections.

Let us now recall the membrane and bending energy as defined in [78, 77, 79].

**Membrane energy** As already explained, the membrane energy measures changes in the first fundamental forms of the undeformed mid-surface  $\bar{\mathcal{M}}$  and the deformed mid-surface  $\phi(\bar{\mathcal{M}}) = \mathcal{M}$ . This can be captured by a field  $A = A[\phi]$  acting on the tangent space of  $\bar{\mathcal{M}}$ . Assume that  $\bar{\mathcal{M}} = \bar{\mathcal{M}}[\bar{y}]$  and  $\mathcal{M} = \mathcal{M}[y]$ , i.e.  $\bar{y}$  is a parameterization of  $\bar{\mathcal{M}}$  and  $y$  is a parameterization of  $\mathcal{M}$  (see Chapter 2). Furthermore, assume we have a point  $\bar{p} = \bar{y}(\bar{\xi})$  with  $\bar{\xi} \in \bar{\Omega}$  where  $\bar{\Omega}$  denotes the corresponding domain manifold of  $\bar{\mathcal{M}}$ , i.e.  $\bar{y} : \bar{\Omega} \rightarrow \bar{\mathcal{M}}$ . Then, the deformation map  $\phi$  maps  $\bar{p}$  to a point  $p = y(\xi) \in \mathcal{M}$  with  $\xi \in \Omega$  (analogously to the previous definition). Then, for tangent vectors  $\bar{U} = D\bar{y}u, \bar{W} = D\bar{y}w \in \mathcal{T}_{\bar{p}}\bar{\mathcal{M}}$  and  $U = Dy u, W = Dy w \in \mathcal{T}_p\mathcal{M}$  for the same  $u, w \in \mathbb{R}^2$  and the corresponding matrices of the first fundamental forms  $\bar{g} = D\bar{y}^T D\bar{y}$  respectively  $g = Dy^T Dy$  we have the following relation

$$g_{\bar{p}}(A_{\bar{p}}\bar{U}, \bar{W}) = g_{\phi(\bar{p})}(D\phi(\bar{V}), D\phi(\bar{W})) = g_p(U, W)$$

where

$$A_{\bar{p}} = D\bar{y}\bar{g}^{-1}A_{\bar{\xi}}D\bar{y}^T \in \mathbb{R}^{3 \times 3}$$

and

$$A_{\bar{\xi}} = g \cdot \bar{g}^{-1} \in \mathbb{R}^{2 \times 2}$$

such that

$$g_{\bar{\xi}}(A_{\bar{\xi}}u, w) = g_{\xi}(u, w).$$

Finally, the membrane energy for all points  $\bar{p}$  of the undeformed mid-surface  $\bar{\mathcal{M}}$  to points  $p = \phi(\bar{p})$  of the deformed mid-surface  $\phi(\bar{\mathcal{M}}) = \mathcal{M}$  reads

$$\begin{aligned} W_{\text{mem}}[\phi] &= W_{\text{mem}}[\bar{\mathcal{M}}, \mathcal{M}] = W_{\text{mem}}(A_{\bar{p}}) = \frac{\mu}{2} \text{tr} A_{\bar{p}} + \frac{\lambda}{4} \det A_{\bar{p}} - \frac{2\mu + \lambda}{4} \log \det A_{\bar{p}} - \mu - \frac{\lambda}{4} \\ &= \frac{\mu}{2} \text{tr} A_{\bar{\xi}} + \frac{\lambda}{4} \det A_{\bar{\xi}} - \frac{2\mu + \lambda}{4} \log \det A_{\bar{\xi}} - \mu - \frac{\lambda}{4}. \end{aligned}$$

Here,  $\lambda$  and  $\mu$  are the Lamé constants (see [78] for details) and  $\text{tr}$  and  $\det$  denote the trace and the determinant of  $A_{\bar{p}}$  respectively  $A_{\bar{\xi}}$  where  $\det A_{\bar{p}}$  describes area distortion, while  $\text{tr} A_{\bar{p}}$  measures length distortion. The function  $W_{\text{mem}}(A)$  is *rigid body motion invariant* and the identity map, i.e. no deformation at all, is the minimizer. The  $\log \det A_{\bar{p}}$  term penalizes material compression, which prevents degeneration of triangles in the discrete setting. For the actual computation we will use  $A_{\bar{\xi}}$  because it is easier to compute and faster in the discrete setting.

**Bending energy** In contrast to the membrane energy, we measure bending by the so-called relative Weingarten map. Therefore, we consider a matrix  $Q[\phi] = Q_{\bar{p}} = \bar{K}_{\bar{p}} - K_{\bar{p}}$  as difference of linear operators acting on tangential vectors on  $\bar{\mathcal{M}}$  (we assume the same notation as for the membrane energy). Here, the matrices  $\bar{K}_{\bar{p}}$  and  $K_{\bar{p}}$  are the shape operators on  $\bar{\mathcal{M}}$  respectively the pulled-back shape operator on  $\mathcal{M}$  to  $\bar{\mathcal{M}}$  defined by

$$g_{\bar{p}}(\bar{K}_{\bar{p}}\bar{U}, \bar{W}) = b_p(\bar{U}, \bar{W})$$

and

$$g_{\bar{p}}(K_{\bar{p}}\bar{U}, \bar{W}) = b_{\phi(\bar{p})}(D\phi(\bar{U}), D\phi(\bar{W})) = b_p(U, W).$$

In Chapter 2 we have seen that

$$\bar{K}_{\bar{p}} = \mathbf{s}_{\bar{p}} = D\bar{x}\bar{g}^{-1}\mathbf{s}_{\bar{\xi}}D\bar{x}^T \in \mathbb{R}^{3 \times 3}$$

and

$$K_{\bar{p}} = D\bar{x}\bar{g}^{-1}b\bar{g}^{-1}D\bar{x}^T \in \mathbb{R}^{3 \times 3}.$$

Then

$$Q[\phi] = Q_{\bar{p}} = \bar{K}_{\bar{p}} - K_{\bar{p}} = D\bar{x}\bar{g}^{-1}(\bar{b} - b)\bar{g}^{-1}D\bar{x}^T = D\bar{x}\bar{g}^{-1}Q_{\bar{\xi}}D\bar{x}^T$$

where

$$Q_{\bar{\xi}} = (\bar{b} - b)\bar{g}^{-1}$$

where we call  $Q_{\bar{p}}$  denotes the *embedded relative Weingarten map* and  $Q_{\bar{\xi}}$  the relative Weingarten map. Since,  $Q_{\bar{\xi}}$  is a parameterization-dependent and  $Q_{\bar{p}}$  is a parameterization-independent linear map, we have to define all continuous objects with respect to  $Q_{\bar{p}}$ . This is crucial for our discrete setting because we will use the mid-edge quadrature weights developed in Chapter 3. The bending energy is

the Frobenius norm of the embedded relative Weingarten map, i.e.

$$W_{\text{bend}}(Q[\phi]) = W_{\text{bend}}(Q_{\bar{p}}) = \text{tr}(Q_{\bar{p}}^T Q_{\bar{p}}) = \text{tr}(Q_{\bar{\xi}} \cdot Q_{\bar{\xi}}).$$

For the last equality we refer again to Chapter 2.

**Discretization with subdivision surfaces** Now we can plug all together and obtain the elastic energy

$$\begin{aligned} \mathcal{W}[\phi] &= \mathcal{W}[\bar{\mathcal{M}}, \mathcal{M}] = \delta \int_{\bar{\mathcal{M}}} W_{\text{mem}}[\phi] \, da + \delta^3 \int_{\bar{\mathcal{M}}} W_{\text{bend}}[\phi] \, da \\ &= \delta \int_{\bar{\Omega}} \left( \frac{\mu}{2} \text{tr}(\bar{g}^{-1} g) + \frac{\lambda}{4} \det(\bar{g}^{-1} g) - \frac{2\mu + \lambda}{4} \log \det(\bar{g}^{-1} g) - \mu - \frac{\lambda}{4} \right) \sqrt{\det \bar{g}} \, d\xi \\ &\quad + \delta^3 \int_{\bar{\Omega}} \text{tr}((\bar{b} - b) \bar{g}^{-1} \cdot (\bar{b} - b) \bar{g}^{-1}) \sqrt{\det \bar{g}} \, d\xi \end{aligned} \quad (5.8)$$

which measures the deformation  $\phi^\delta$  from an undeformed shell  $\bar{\mathcal{M}}^\delta$  to a deformed shell  $\mathcal{M}^\delta$  by comparing the membrane and bending energy of the corresponding mid-surfaces  $\bar{\mathcal{M}}$  and  $\mathcal{M}$  scaled by the thickness  $\delta$ . Furthermore, the  $\delta$ -scaling is not artificial but relies on analytical results obtain in [62, 63, 61, 93] by  $\Gamma$ -convergence for  $\delta \rightarrow 0$ . Let us first remark that the energy  $\mathcal{W}$  in 5.8 is *rigid body motion invariant*, i.e. for all points  $\bar{p} \in \bar{\mathcal{M}}$  and a coordination transformation  $\tilde{p} = R\bar{p} + b$  where  $R \in SO(3)$  we obtain

$$W_{\text{mem}}(A_{\tilde{p}}) = W_{\text{mem}}(A_{\bar{p}})$$

and

$$W_{\text{bend}}(A_{\tilde{p}}) = W_{\text{bend}}(A_{\bar{p}}).$$

The central ingredient of the proposed framework is the approximation of the squared Riemannian distance by a functional (5.3) that reflects the structure of the underlying shape space  $\mathcal{S}$ . For the energy functional we approximated the path energy (5.5) as the second central ingredient and in (5.8) we have defined the continuous elastic energy to compare two shells of thickness  $\delta$ .

Therefore, we will only spatially discretize the path energy (5.5) by discretizing the elastic energy (5.8) as follows. Assume that we have two subdivision surfaces  $Y_{k-1}$  and  $Y_k$  given (see Chapter 2 and 3) then we write the discretized elastic energy  $\mathbf{W}$  of  $\mathcal{W}$  as

$$\begin{aligned} \mathbf{W}[Y_{k-1}, Y_k] &= \sum_{k \in \mathcal{I}_c} \int_{\Delta} \left( \frac{\mu}{2} \text{tr}(G^{-1}[Y_{k-1}(\xi, k)] G[Y_k(\xi, k)]) + \frac{\lambda}{4} \det(G^{-1}[Y_{k-1}(\xi, k)] G[Y_k(\xi, k)]) \right. \\ &\quad \left. - \frac{2\mu + \lambda}{4} \log \det(G^{-1}[Y_{k-1}(\xi, k)] G[Y_k(\xi, k)]) - \mu - \frac{\lambda}{4} \right) \sqrt{\det G[Y_{k-1}(\xi, k)]} \, d\xi \\ &\quad + \delta^2 \int_{\Delta} \text{tr} \left( ((B[Y_{k-1}(\xi, k)] - B[Y_k(\xi, k)]) G^{-1}[Y_{k-1}(\xi, k)])^2 \right) \sqrt{\det G[Y_{k-1}(\xi, k)]} \, d\xi. \end{aligned} \quad (5.9)$$

In this definition we multiplied  $\mathcal{W}$  by  $\frac{1}{\delta}$  which is numerically more stable in the later implementation.

Let  $Y_A, Y_B \in \mathcal{S}_h$  be two given subdivision surfaces, then a sequence  $Y_0, \dots, Y_K \in \mathcal{S}_h$  of subdivi-



sion surfaces with  $Y_0 = Y_A$ ,  $Y_K = Y_B$  and  $Y_k \in \mathcal{S}_h$  has a path energy

$$\mathbf{E}[Y_0, \dots, Y_K] = K \sum_{k=1}^K \mathbf{W}[Y_{k-1}, Y_k],$$

where  $\mathcal{S}_h = \mathcal{V}_h$  denotes the *subdivision shell space* (see equation (4.15) for a definition of  $\mathcal{V}_h$ ), i.e. all  $Y_k \in \mathcal{S}_h$  have a subdivision parameterization and describe a mid-surface of a shell with thickness  $\delta$ . All other definitions such as discrete path length, discrete geodesic interpolation, discrete logarithm and exponential map, discrete parallel transport, discrete Bézier curve, discrete B-spline and discrete cardinal splines follow from this definition.

## 5.4. Numerical Implementation

In this section we describe the implementation of the previously defined concept with subdivision surfaces. The main ingredient is the minimization of the elastic path energy using Newton's method for the corresponding Lagrangian incorporating rigid body motions.

**Computing discrete geodesics** Now, we want to describe how we compute a discrete geodesic in the space of subdivision shells. Therefore, consider a  $K + 1$ -tuple of subdivision surfaces  $Y_k = \sum_{i \in \mathcal{I}_v} (\bar{Y}_k)_i \Phi_i \in \mathcal{S}_h$  with  $k = 0, \dots, K$ . Here, we used the notation  $\bar{Y}_k = (\bar{Y}_k^1, \bar{Y}_k^2, \bar{Y}_k^3)^T \in \mathbb{R}^{3 \times |\mathcal{I}_v|}$  and  $(\bar{Y}_k)_i = ((\bar{Y}_k)_i^1, (\bar{Y}_k)_i^2, (\bar{Y}_k)_i^3)^T \in \mathbb{R}^3$  (for details see Chapter 2). Furthermore, we assume we have two given shapes, the first shape  $Y_0 = Y_A \in \mathcal{S}_h$  and the end shape  $Y_K = Y_B \in \mathcal{S}_h$ , and that *both control meshes have the same connectivity*. The task is now to compute all intermediate shapes  $(Y_1, \dots, Y_{K-1}) \in \underbrace{\mathcal{S}_h \times \dots \times \mathcal{S}_h}_{K-1}$  (again, all with the same connectivity as the input shapes  $Y_A$  and  $Y_B$ ) such that  $(Y_0, Y_1, \dots, Y_{K-1}, Y_K)$  is a minimizer of the following optimization problem

$$\min_{(Y_0, \dots, Y_K)} \mathbf{E}[(Y_0, \dots, Y_K)] = K \sum_{k=1}^K \mathbf{W}[Y_{k-1}, Y_k] \quad (5.10a)$$

$$\text{s.t.} \quad Y_0 = Y_A, \quad Y_K = Y_B, \quad (5.10b)$$

$$\text{(translation)} \quad \int_{\mathcal{M}[Y_0]} Y_0 - Y_k \, da = 0, \quad k = 1, \dots, K-1, \quad (5.10c)$$

$$\text{(rotation)} \quad \int_{\mathcal{M}[Y_0]} Y_0 \times Y_k \, da = 0, \quad k = 1, \dots, K-1. \quad (5.10d)$$

The constraints (5.10c) and (5.10d) reflect that the energy  $\mathcal{W}$  is invariant under rigid body motion, i.e. translation and rotation. We write the minimization problem (5.10) to the associate *Lagrangian function*

$$\begin{aligned} \mathbf{L}(\mathbf{Y}, \boldsymbol{\Lambda}, \boldsymbol{\Theta}) &= K \sum_{k=1}^K \mathbf{W}[Y_{k-1}, Y_k] + \sum_{k=1}^{K-1} \sum_{i=1}^3 \int_{\mathcal{M}[Y_0]} \lambda_k^i (Y_0^i - Y_k^i) \, da \\ &\quad + \sum_{k=1}^{K-1} \int_{\mathcal{M}[Y_0]} \theta_k^1 (Y_0^1 Y_k^2 - Y_0^2 Y_k^1) + \theta_k^2 (Y_0^2 Y_k^3 - Y_0^3 Y_k^2) + \theta_k^3 (Y_0^3 Y_k^1 - Y_0^1 Y_k^3) \, da, \end{aligned}$$

## 5. Geodesics, Bézier Curves and Splines in Subdivision Shell Space

where  $\mathbf{Y} = (\bar{\theta}_1, \dots, \bar{\theta}_{K-1}) \in \mathbb{R}^{3 \cdot (K-1) \cdot |\mathcal{I}_v|}$  denotes the vector of the control vectors  $\bar{Y}_k \in \mathbb{R}^{3 \cdot |\mathcal{I}_v|}$  of the subdivision surface  $Y_k$ ,  $\mathbf{\Lambda} = (\bar{\lambda}_1, \dots, \bar{\lambda}_{K-1}) \in \mathbb{R}^{3 \cdot (K-1)}$  is the vector of Lagrange multipliers for the translation constraints and  $\mathbf{\Theta} = (\bar{Y}_1, \dots, \bar{Y}_{K-1}) \in \mathbb{R}^{3 \cdot (K-1)}$  is the vector of Lagrange multipliers for the rotation constraints. Here, we used that  $\bar{\theta}_k = (\theta_k^1 \ \theta_k^2 \ \theta_k^3)^T \in \mathbb{R}^3$  and  $\bar{\lambda}_k = (\lambda_k^1 \ \lambda_k^2 \ \lambda_k^3)^T \in \mathbb{R}^3$ . Then, the necessary condition that  $(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})$  is a minimizer of (5.10) reads

$$DL(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta}) = \begin{pmatrix} \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{Y}} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{\Lambda}} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{\Theta}} \end{pmatrix} = 0,$$

where

$$\frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{Y}} = \begin{pmatrix} \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^1} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^2} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^3} \\ \vdots \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^1} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^2} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^3} \end{pmatrix}, \quad \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{\Lambda}} = \begin{pmatrix} \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \lambda_1^1} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \lambda_1^2} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \lambda_1^3} \\ \vdots \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \lambda_{K-1}^1} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \lambda_{K-1}^2} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \lambda_{K-1}^3} \end{pmatrix} \quad \text{and} \quad \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{\Theta}} = \begin{pmatrix} \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \theta_1^1} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \theta_1^2} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \theta_1^3} \\ \vdots \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \theta_{K-1}^1} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \theta_{K-1}^2} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \theta_{K-1}^3} \end{pmatrix}.$$

To set up a Newton method we further need the Hessian of L

$$D^2L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta}) = \begin{pmatrix} \frac{\partial^2 L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial^2 \mathbf{Y}} & \frac{\partial^2 L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{\Lambda} \partial \mathbf{Y}} & \frac{\partial^2 L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{\Theta} \partial \mathbf{Y}} \\ \frac{\partial^2 L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{Y} \partial \mathbf{\Lambda}} & \mathbf{0} & \mathbf{0} \\ \frac{\partial^2 L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{Y} \partial \mathbf{\Theta}} & \mathbf{0} & \mathbf{0} \end{pmatrix},$$

where  $\mathbf{0}$  denotes the corresponding zero matrix.

$$\frac{\partial^2 L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial^2 \mathbf{Y}} = \begin{pmatrix} \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^1 \partial Y_1^1} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^2 \partial Y_1^1} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^3 \partial Y_1^1} & \cdots & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^1 \partial Y_1^1} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^2 \partial Y_1^1} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^3 \partial Y_1^1} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^1 \partial Y_1^2} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^2 \partial Y_1^2} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^3 \partial Y_1^2} & \cdots & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^1 \partial Y_1^2} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^2 \partial Y_1^2} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^3 \partial Y_1^2} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^1 \partial Y_1^3} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^2 \partial Y_1^3} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^3 \partial Y_1^3} & \cdots & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^1 \partial Y_1^3} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^2 \partial Y_1^3} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^3 \partial Y_1^3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^1 \partial Y_{K-1}^1} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^2 \partial Y_{K-1}^1} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^3 \partial Y_{K-1}^1} & \cdots & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^1 \partial Y_{K-1}^1} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^2 \partial Y_{K-1}^1} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^3 \partial Y_{K-1}^1} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^1 \partial Y_{K-1}^2} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^2 \partial Y_{K-1}^2} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^3 \partial Y_{K-1}^2} & \cdots & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^1 \partial Y_{K-1}^2} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^2 \partial Y_{K-1}^2} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^3 \partial Y_{K-1}^2} \\ \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^1 \partial Y_{K-1}^3} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^2 \partial Y_{K-1}^3} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_1^3 \partial Y_{K-1}^3} & \cdots & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^1 \partial Y_{K-1}^3} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^2 \partial Y_{K-1}^3} & \frac{\partial L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial Y_{K-1}^3 \partial Y_{K-1}^3} \end{pmatrix}$$

and analogously for  $\frac{\partial^2 L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{\Lambda} \partial \mathbf{Y}}$  and  $\frac{\partial^2 L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{\Theta} \partial \mathbf{Y}}$ . Furthermore,  $\frac{\partial^2 L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{Y} \partial \mathbf{\Lambda}} = \left( \frac{\partial^2 L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{\Lambda} \partial \mathbf{Y}} \right)^T$  and  $\frac{\partial^2 L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{Y} \partial \mathbf{\Theta}} = \left( \frac{\partial^2 L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta})}{\partial \mathbf{\Theta} \partial \mathbf{Y}} \right)^T$ .

For given initial start vectors  $(\mathbf{Y}^0, \mathbf{\Lambda}^0, \mathbf{\Theta}^0)$  we solve for every Newton step  $\ell \geq 0$  the linear system

$$D^2 L(\mathbf{Y}^\ell, \mathbf{\Lambda}^\ell, \mathbf{\Theta}^\ell) \Delta(\mathbf{Y}^{\ell+1}, \mathbf{\Lambda}^{\ell+1}, \mathbf{\Theta}^{\ell+1}) = -DL(\mathbf{Y}^\ell, \mathbf{\Lambda}^\ell, \mathbf{\Theta}^\ell),$$

compute the next Newton steps by

$$(\mathbf{Y}^{\ell+1}, \mathbf{\Lambda}^{\ell+1}, \mathbf{\Theta}^{\ell+1}) = (\mathbf{Y}^\ell, \mathbf{\Lambda}^\ell, \mathbf{\Theta}^\ell) + \eta \Delta(\mathbf{Y}^{\ell+1}, \mathbf{\Lambda}^{\ell+1}, \mathbf{\Theta}^{\ell+1})$$

until

$$\|\Delta(\mathbf{Y}^{\ell+1}, \mathbf{\Lambda}^{\ell+1}, \mathbf{\Theta}^{\ell+1})\| < \rho$$

where  $\eta \in [0, 1]$  is the step size computed by ([125], p.129) and  $\rho \ll 1$  is given. The crucial step in the implementation is the choice of a starting shapes  $\mathbf{Y}^0$  ( $\mathbf{\Lambda}^\ell, \mathbf{\Theta}^\ell$  can be chosen to be zero vectors in order to make the iteration work). One possibility is a linear blend between the input shapes  $Y_A$  and  $Y_B$ , but as Heeren [78] demonstrated self-penetration is a problem for complex shapes. Therefore, we consider the following example. Given two points  $Y_0 = Y_A, Y_2 = Y_B \in \mathbb{R}^2$ , an unknown point  $Y_1 \in \mathbb{R}^2$  and the functional

$$F[Y_1] = \frac{1}{\tau_1} \mathbf{W}[Y_0, Y_1] + \frac{1}{\tau_2} \mathbf{W}[Y_1, Y_2],$$

where

$$\mathbf{W}[Y_i, Y_j] = |Y_i - Y_j|^2 = (Y_i^1 - Y_j^1)^2 + (Y_i^2 - Y_j^2)^2,$$

i.e.  $\mathbf{W}$  is the squared Euclidean distance. Then, the optimality condition reads

$$\begin{aligned} F_{,k}[Y_1] &= \partial_{Y_1^k} \left( \frac{1}{\tau_1} \mathbf{W}[Y_0, Y_1] + \frac{1}{\tau_2} \mathbf{W}[Y_1, Y_2] \right) \\ &= -\frac{2}{\tau_1} (Y_0^k - Y_1^k) + \frac{2}{\tau_2} (Y_1^k - Y_2^k) = 0, \end{aligned}$$

which has a unique solution

$$Y_1^k = \frac{1}{\tau_1 + \tau_2} (\tau_2 Y_0^k + \tau_1 Y_2^k).$$

For the special case  $\tau_1 = t, \tau_2 = 1 - t$  with  $t \in (0, 1)$  we have

$$Y_1^k = (1 - t)Y_0^k + tY_2^k,$$

i.e.  $Y_1$  lies on the straight line between  $Y_0$  and  $Y_2$ .

Based on this observation we compute initial values for the Newton method by solving  $K - 1$

smaller problems. Consider a 3-point geodesic, i.e.  $K = 2$ , and the functional

$$F^\tau[Y_1] = \frac{1}{\tau} \mathbf{W}[Y_0, Y_1] + \frac{1}{1-\tau} \mathbf{W}[Y_1, Y_2]$$

where  $\mathbf{W}$  is again defined as in (5.9). The basic idea is now to compute a  $K + 1$ -geodesic, i.e.  $(Y_0, \dots, Y_K)$  with  $Y_0 = Y_A$  and  $Y_K = Y_B$ , the initial shapes  $Y_k^0$  for  $\mathbf{Y}^0$  as the minimizer of  $\arg \min F^\tau[Y_1] = Y_k^0$  for  $\tau = \frac{k}{K}$ . Let us be more precise, we want to optimize the corresponding Lagrangian

$$\begin{aligned} L^\tau(\mathbf{Y}, \boldsymbol{\Lambda}, \boldsymbol{\Theta}) &= F^\tau[Y_1] + \sum_{i=1}^3 \int_{\mathcal{M}[Y_0]} \lambda_1^i (Y_0^i - Y_1^i) da \\ &+ \int_{\mathcal{M}[Y_0]} \theta_1^1 (Y_0^1 Y_1^2 - Y_0^2 Y_1^1) + \theta_1^2 (Y_0^2 Y_1^3 - Y_0^3 Y_1^2) + \theta_1^3 (Y_0^3 Y_1^1 - Y_0^1 Y_1^3) da \end{aligned}$$

by computing the root

$$\nabla L^\tau(\mathbf{Y}, \boldsymbol{\Lambda}, \boldsymbol{\Theta}) = \begin{pmatrix} F_{,1}^\tau[Y_1] - \lambda_1^1 \mathbf{M}_\perp - \theta_1^1 \mathbf{M}_0^2 + \theta_1^3 \mathbf{M}_0^3 \\ F_{,2}^\tau[Y_1] - \lambda_1^2 \mathbf{M}_\perp - \theta_1^2 \mathbf{M}_0^3 + \theta_1^1 \mathbf{M}_0^1 \\ F_{,3}^\tau[Y_1] - \lambda_1^3 \mathbf{M}_\perp - \theta_1^3 \mathbf{M}_0^1 + \theta_1^2 \mathbf{M}_0^2 \\ \mathbb{1}^T \mathbf{M} (\bar{Y}_0^1 - \bar{Y}_1^1) \\ \mathbb{1}^T \mathbf{M} (\bar{Y}_0^2 - \bar{Y}_1^2) \\ \mathbb{1}^T \mathbf{M} (\bar{Y}_0^3 - \bar{Y}_1^3) \\ (\bar{Y}_0^1)^T \mathbf{M} \bar{Y}_1^2 - (\bar{Y}_0^2)^T \mathbf{M} \bar{Y}_1^1 \\ (\bar{Y}_0^2)^T \mathbf{M} \bar{Y}_1^3 - (\bar{Y}_0^3)^T \mathbf{M} \bar{Y}_1^2 \\ (\bar{Y}_0^3)^T \mathbf{M} \bar{Y}_1^1 - (\bar{Y}_0^1)^T \mathbf{M} \bar{Y}_1^3 \end{pmatrix},$$

where

$$F_{,i}^\tau[Y_1] = \frac{\partial}{\partial \bar{Y}_1^i} \left( \frac{1}{\tau} \mathbf{W}[Y_0, Y_1] + \frac{1}{1-\tau} \mathbf{W}[Y_1, Y_2] \right),$$

and  $\mathbf{M}_\perp = \mathbf{M} \cdot \mathbb{1}$ ,  $\mathbf{M}_0^i = \mathbf{M} \cdot Y_0^i$ ,  $\mathbb{1}$  is the vector with ones and  $\mathbf{M}$  denotes the subdivision mass matrix (see definition (3.10)). To set up a Newton method we compute the Hessian by

$$\nabla^2 L^\tau(\mathbf{Y}, \boldsymbol{\Lambda}, \boldsymbol{\Theta}) = \begin{pmatrix} F_{,11}^\tau[Y_1] & F_{,12}^\tau[Y_1] & F_{,13}^\tau[Y_1] & -\mathbf{M}_\perp & 0 & 0 & -\mathbf{M}_0^2 & 0 & \mathbf{M}_0^3 \\ F_{,21}^\tau[Y_1] & F_{,22}^\tau[Y_1] & F_{,23}^\tau[Y_1] & 0 & -\mathbf{M}_\perp & 0 & \mathbf{M}_0^1 & -\mathbf{M}_0^3 & 0 \\ F_{,31}^\tau[Y_1] & F_{,32}^\tau[Y_1] & F_{,33}^\tau[Y_1] & 0 & 0 & -\mathbf{M}_\perp & 0 & \mathbf{M}_0^2 & -\mathbf{M}_0^1 \\ -(\mathbf{M}_\perp)^T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -(\mathbf{M}_\perp)^T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -(\mathbf{M}_\perp)^T & 0 & 0 & 0 & 0 & 0 & 0 \\ -(\mathbf{M}_0^2)^T & (\mathbf{M}_0^1)^T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -(\mathbf{M}_0^3)^T & (\mathbf{M}_0^2)^T & 0 & 0 & 0 & 0 & 0 & 0 \\ (\mathbf{M}_0^3)^T & 0 & -(\mathbf{M}_0^1)^T & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

where

$$F_{,ij}^\tau[Y_1] = \frac{\partial^2}{\partial \bar{Y}_1^j \partial \bar{Y}_1^i} \left( \frac{1}{\tau} \mathbf{W}[Y_0, Y_1] + \frac{1}{1-\tau} \mathbf{W}[Y_1, Y_2] \right).$$

Again, the root  $Y_1$  of  $L^\tau$  for  $\tau = \frac{k}{K}$  is then used as the starting value  $Y_k^0$  for the optimization of  $L$  for all  $K$ . For a detailed derivation of the derivatives of  $\mathbf{W}$  with respect to the subdivision control points see Appendix A.4. Let us remark, that  $\mathbf{I}^K(Y_A, Y_B, k) = Y_k$  is the  $k$ -th component of the solution  $\mathbf{Y} = (Y_1, \dots, Y_{K-1})$ . Furthermore, the computation of  $\mathbf{B}^K$ ,  $\mathbf{QBS}^{K,M}$ , and  $\mathbf{CBS}^{K,M}$  is just the computation of recursively defined discrete geodesics which can be found in Section 5.1 and 5.2.

**Discrete exponential map** Let us now shortly describe the computation of the discrete exponential map  $\mathbf{EXP}_{\bar{Y}_0}^K[\Xi_0]$  for a given subdivision surface  $Y_0$  and a displacement subdivision function  $\Xi_0$ , i.e. corresponding displacement vector  $\bar{\Xi}_0$  with respect to the subdivision control points  $\bar{Y}_0$ . In system (5.7) we have seen that everything reduces to the computation of  $\mathbf{EXP}_{\bar{Y}_0}^2[\Xi_0]$ . Analogously to the spatial continuous case we want to compute the root  $Y_2 \in \mathcal{S}_h$  of

$$\begin{pmatrix} E_1[Y_2] \\ E_2[Y_2] \\ E_3[Y_2] \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial \bar{Y}_1^1} (2\mathbf{W}[Y_0, Y_1] + 2\mathbf{W}[Y_1, Y_2]) \\ \frac{\partial}{\partial \bar{Y}_1^2} (2\mathbf{W}[Y_0, Y_1] + 2\mathbf{W}[Y_1, Y_2]) \\ \frac{\partial}{\partial \bar{Y}_1^3} (2\mathbf{W}[Y_0, Y_1] + 2\mathbf{W}[Y_1, Y_2]) \end{pmatrix}$$

where  $Y_1 = Y_0 + \Xi_0$  is defined via the control points  $\bar{Y}_1 = \bar{Y}_0 + \bar{\Xi}_0$ . As for the discrete geodesics we need to incorporate rigid body motions such that we will compute the roots of the following gradient of the Lagrangian

$$DL(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta}) = \begin{pmatrix} E_1[Y_2] - \lambda_1^1 \mathbf{M} \cdot \mathbb{1} - \theta_1^1 \mathbf{M}_0^2 + \theta_1^3 \mathbf{M}_0^3 \\ E_2[Y_2] - \lambda_1^2 \mathbf{M} \cdot \mathbb{1} - \theta_1^2 \mathbf{M}_0^3 + \theta_1^1 \mathbf{M}_0^1 \\ E_3[Y_2] - \lambda_1^3 \mathbf{M} \cdot \mathbb{1} - \theta_1^3 \mathbf{M}_0^1 + \theta_1^2 \mathbf{M}_0^2 \\ \mathbb{1}^T \mathbf{M} (\bar{Y}_0^1 - \bar{Y}_2^1) \\ \mathbb{1}^T \mathbf{M} (\bar{Y}_0^2 - \bar{Y}_2^2) \\ \mathbb{1}^T \mathbf{M} (\bar{Y}_0^3 - \bar{Y}_2^3) \\ (\bar{Y}_0^1)^T \mathbf{M} \bar{Y}_2^2 - (\bar{Y}_0^2)^T \mathbf{M} \bar{Y}_2^1 \\ (\bar{Y}_0^2)^T \mathbf{M} \bar{Y}_2^3 - (\bar{Y}_0^3)^T \mathbf{M} \bar{Y}_2^2 \\ (\bar{Y}_0^3)^T \mathbf{M} \bar{Y}_2^1 - (\bar{Y}_0^1)^T \mathbf{M} \bar{Y}_2^3 \end{pmatrix}.$$

Again, to set up the Newton method we need the Hessian

$$D^2L(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Theta}) = \begin{pmatrix} E_{1,1}[Y_2] & E_{1,2}[Y_2] & E_{1,3}[Y_2] & -\mathbf{M}_1 & 0 & 0 & -\mathbf{M}_0^2 & 0 & \mathbf{M}_0^3 \\ E_{2,1}[Y_2] & E_{2,2}[Y_2] & E_{2,3}[Y_2] & 0 & -\mathbf{M}_1 & 0 & \mathbf{M}_0^1 & -\mathbf{M}_0^3 & 0 \\ E_{3,1}[Y_2] & E_{3,2}[Y_2] & E_{3,3}[Y_2] & 0 & 0 & -\mathbf{M}_1 & 0 & \mathbf{M}_0^2 & -\mathbf{M}_0^1 \\ -(\mathbf{M}_1)^T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -(\mathbf{M}_1)^T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -(\mathbf{M}_1)^T & 0 & 0 & 0 & 0 & 0 & 0 \\ -(\mathbf{M}_0^2)^T & (\mathbf{M}_0^1)^T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -(\mathbf{M}_0^3)^T & (\mathbf{M}_0^2)^T & 0 & 0 & 0 & 0 & 0 & 0 \\ (\mathbf{M}_0^3)^T & 0 & -(\mathbf{M}_0^1)^T & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

where

$$E_{i,j}[Y_2] = \frac{\partial^2}{\partial \bar{Y}_2^j \partial \bar{Y}_1^i} (2\mathbf{W}[Y_0, Y_1] + 2\mathbf{W}[Y_1, Y_2]).$$

Finally, we obtain the solution  $Y_2$  by solving the corresponding system with the Newton iteration described above.

Let us remark that the computation of  $\mathbf{P}^K$  respectively  $\mathbf{CS}_\kappa^K$  is just the recursive computation of discrete geodesics and discrete exponential map (for details see Section 5.1 and 5.2).

**Eigenmodes of the elastic deformation energy** Let us conclude this section by introducing the computation of eigenmodes of the discrete elastic energy  $\mathcal{W}$  respectively  $\mathbf{W}$  which results in a fairly easy tool for animation as demonstrated in Section 5.5. The authors of [79] call the concept behind this *animation without animating*, because it allows an intrinsic animation based on the defined elastic energy  $\mathcal{W}$  respectively  $\mathbf{W}$ . In particular, eigenmodes of  $\mathcal{W}$  have to be understood as eigenmodes of the Hessian of  $\mathcal{W}$

$$\text{Hess } \mathcal{W}[\mathbb{1}](\phi, \psi) = \left( \frac{d}{ds} \left( \frac{d}{dt} \mathcal{W}[y_0, y_0 + t\phi + s\psi] \right)_{t=0} \right)_{s=0},$$

where  $y_0 \in \mathcal{S}$  is a given shape,  $\phi = \mathbb{1}$  denotes the identity map and  $\phi, \psi \in \mathcal{S}$  are test functions. Analogously, in the discrete setting this becomes

$$A = \text{Hess } \mathbf{W}[\mathbb{1}](\Phi, \Psi) = \left( \frac{\partial^2}{\partial^2 Y_1} \mathbf{W}[Y_0, Y_1] \right)_{Y_1=Y_0}$$

for a given shape  $Y_0 \in \mathcal{S}_h$ . Then, the eigenvectors  $\bar{V}_k$  respectively the eigenmodes  $V_k$  can easily be computed using *inverse vector iteration* (see [125]) for the system

$$A\bar{V}_k = \Lambda_k \begin{pmatrix} \mathbf{M} & 0 & 0 \\ 0 & \mathbf{M} & 0 \\ 0 & 0 & \mathbf{M} \end{pmatrix} \bar{V}_k,$$

where  $\Lambda_k$  denotes the eigenvalue corresponding to  $V_k$ . Because  $V_k$  is just the eigenmode of the deformation, the resulting deformed subdivision surface with respect to  $V_k$  reads  $X_k = Y_0 + V_k \in \mathcal{S}_h$  respectively for the control points  $\bar{X}_k = \bar{Y}_0 + \bar{V}_k \in \mathbb{R}^{3 \cdot |\mathcal{I}_v|}$ . Let us remark, that we have to incorporate rigid body motions into the solver if we do not want to have the first six eigenmodes to be the trivial ones, i.e translations and rotations.

## 5.5. Numerical Results

We have implemented the proposed method in C++ and tested it for a set of input shapes. Let us emphasize that all computations were done with the coarse control meshes and that all meshes have at least one element (i.e. triangle) that has more than two extraordinary vertices. To our knowledge, this is the first work that presents results for subdivision surfaces without subdividing the coarse control mesh for analysis in this case (i.e. for meshes with triangles with more than two extraordinary vertices). This is possible due to the mid-edge evaluation process developed in Chapter 3. Furthermore, as discussed

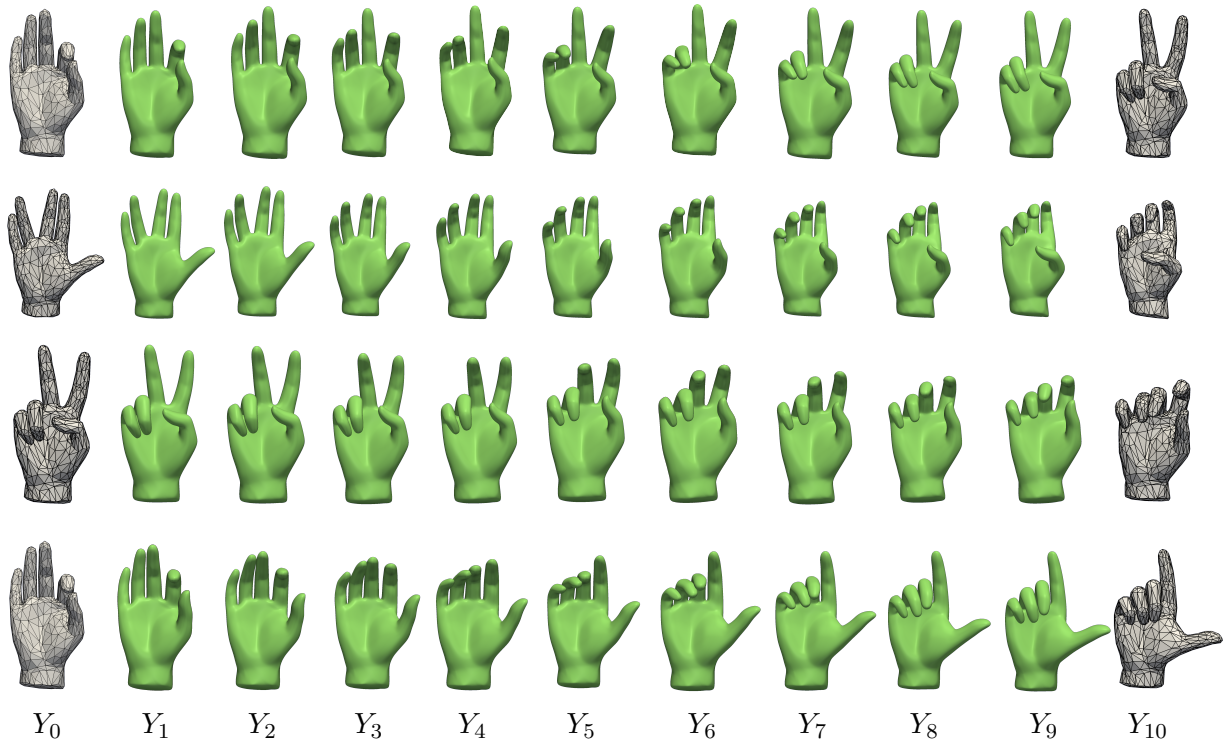


Figure 5.9.: **Discrete Geodesics:** Four different discrete geodesics for different input poses of a hand and  $K = 10$ . The input shapes are visualized as control meshes and the shapes on a discrete geodesic with  $K = 10$  are shown as smooth subdivision surfaces in green. All meshes have the same connectivity and  $|\mathcal{I}_v| = 610$  vertices. Furthermore, all computations were performed with  $\delta = 0.01$ .

in Chapter 2, we consider only closed surfaces, i.e. closed meshes in the discrete setting. Finally, let us remark, that we use a consistent color coding in all pictures. Curves and shapes shown in green imply respectively were computed via discrete geodesics, in pink imply respectively were computed via either discrete exponential maps or discrete parallel transport and in blue imply respectively were computed via discrete Bézier curves including discrete B-splines and discrete cardinal splines.

The fundamental tools for the development of discrete geodesic Bézier curves, B-splines and cardinal splines are discrete geodesics, discrete logarithm and exponential map, the eigenmodes of the elastic deformation energy and the discrete parallel transport developed in [123, 78, 77, 79] and described in detail in Section 5.1. Furthermore, the definition of the underlying shape space, i.e. the shell space defined in Section 5.3, was again taken from [78, 77, 79]. In comparison to [78, 77, 79] we discretized the continuous elastic energy (5.8) and thus the overall concept of discrete geodesic calculus with conforming subdivision isogeometric elements (see Chapter 2 and 3) instead of discretizing the continuous elastic energy with *discrete differential geometry* as done in [78, 77, 79]. We obtain visually similar results as in [78, 77, 79] but for smooth subdivision surfaces with coarse control meshes instead of pure triangle meshes with many degrees of freedom. For instance in Figure 5.9 we depict four discrete geodesics between different poses of a hand with relatively few degrees of freedom ( $|\mathcal{I}_v| = 610$  vertices). In Figure 5.10 we visualize the discrete extrapolation of the motion of an elephant as a result

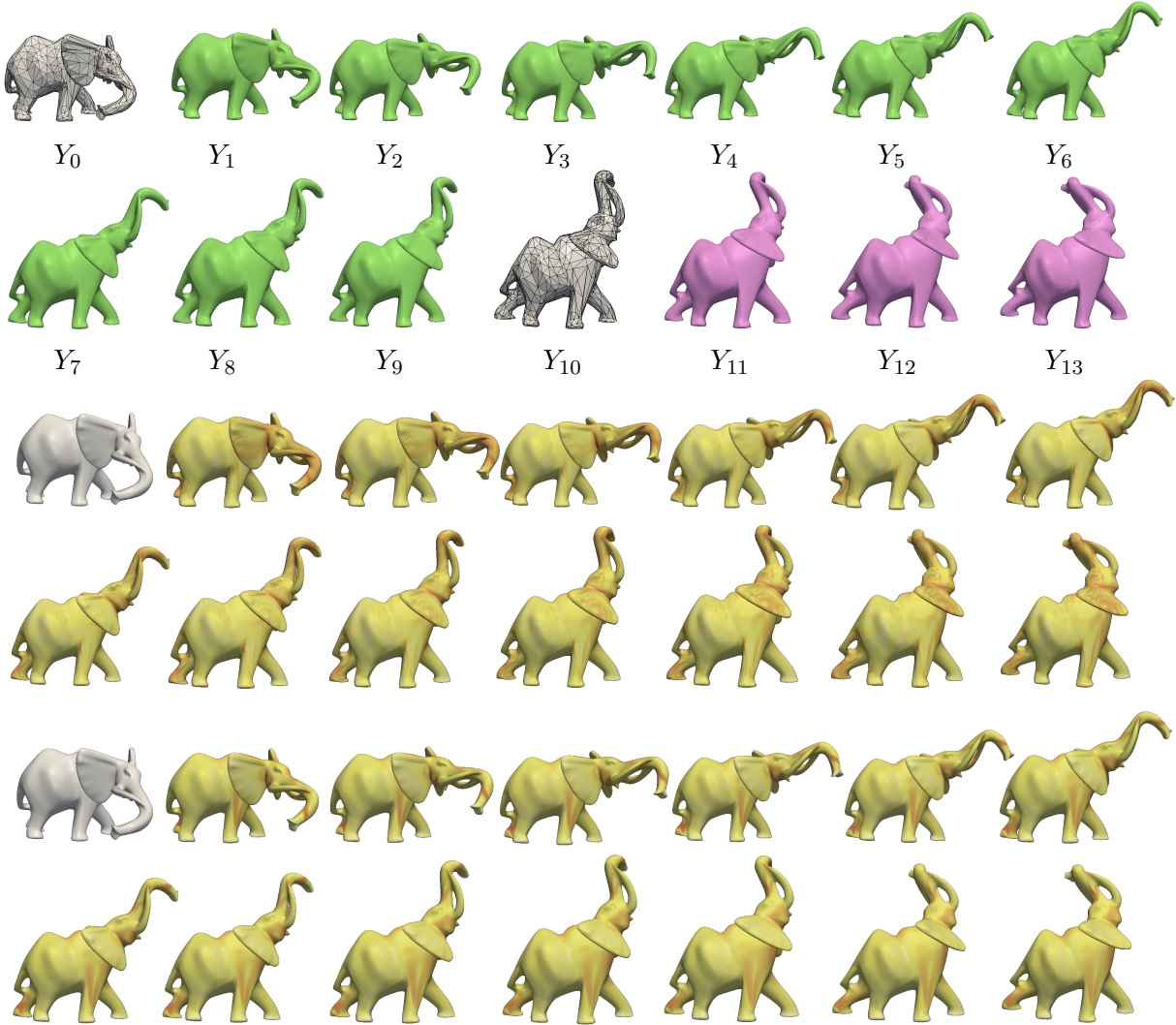


Figure 5.10.: **Discrete Exponential Map:** First, we computed a discrete geodesic between shape  $Y_0$  and  $Y_{10}$  with  $K = 10$  and extrapolated the deformation three times, i.e.  $Y_{11} = \mathbf{EXP}_{Y_0}^2[Y_{10} - Y_9]$ ,  $Y_{12} = \mathbf{EXP}_{Y_{10}}^2[Y_{11} - Y_{10}]$  and  $Y_{13} = \mathbf{EXP}_{Y_{10}}^2[Y_{12} - Y_{11}]$ . The input shapes ( $Y_0$  and  $Y_{10}$ ) in row one respectively two are depicted as control meshes ( $|\mathcal{I}_v| = 1000$ ), while the computed discrete geodesics (the green shapes in row one and two) and the extrapolated shapes (pink shapes in row two) are shown as smooth subdivision surfaces. In the third and fourth row we depict the bending energy densities  $\mathbf{W}_{\text{bend}}[Y_{k-1}, Y_k]$  and in the fifth and sixth row the membrane energy densities  $\mathbf{W}_{\text{mem}}[Y_{k-1}, Y_k]$  each plotted on shape  $Y_k$ . In both cases, the color code from white to yellow to red reflects the strength of the deformations, i.e. from no deformation to strong deformation. The computation was performed with  $\delta = 0.8$ .



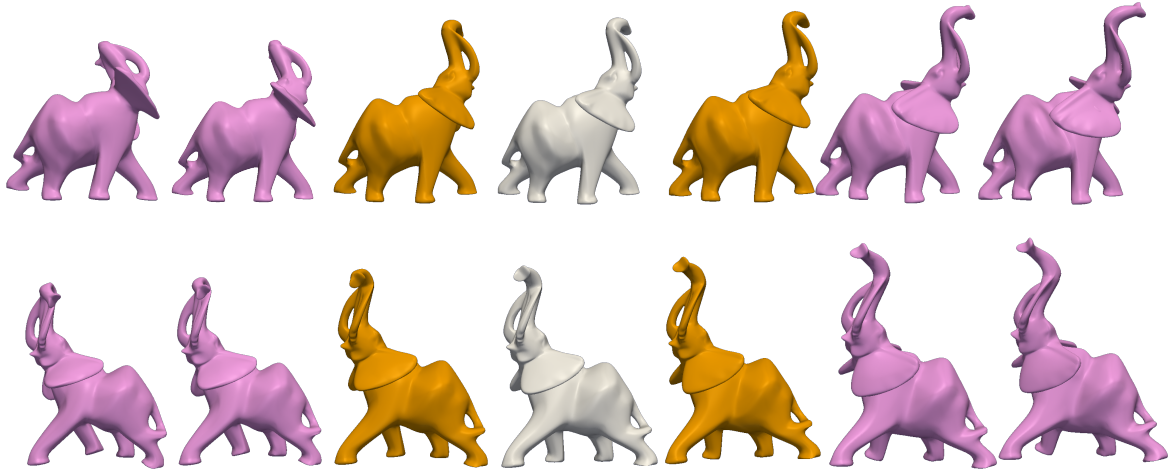


Figure 5.11.: **Eigenmodes of the Elastic Energy  $\mathbf{W}$** : Discrete extrapolation from a shell (depicted in gray, which is  $Y_{10}$  taken from 5.10) along two different initial displacements, given by the first (top) and seventh (bottom) non-trivial eigenmode of the Hessian of  $\mathbf{W}$  where  $\delta = 0.8$ . Besides the input shape, the eigenmode (left and right scaled by  $-1$  respectively  $+1$ ) and in each case two extrapolated shapes.

of a discrete geodesic and in Figure 5.11 a discrete extrapolation of the first and seventh non-trivial eigenmode of the elastic energy  $\mathcal{W}$  respectively  $\mathbf{W}$  in the discrete setting. Finally, we transferred the motion of a cat via discrete parallel transport to a lion with same connectivity in Figure 5.12. Let us remark that we actually never computed the discrete logarithm map and worked instead directly with the meshes from the discrete geodesic respectively discrete exponential map. The reason is simple, all computations were done with respect to rigid body motions, i.e. we never fixed at least two vertices of the meshes. In this case, the discrete logarithm  $\frac{1}{\mathbf{I}}\mathbf{LOG}_{Y_k}[Y_{k+1}] = Y_{k+1} - Y_k$  of two shapes  $Y_k$  and  $Y_{k+1}$  has to be understood with respect to rigid body motions, i.e. with respect to translation and rotation invariance, and not pointwise!

In Figure 5.13 and 5.14 we depict a discrete quadratic Bézier curve with  $K = 10$  respectively a discrete cubic Bézier curve with  $K = 12$ . In both cases the generated discrete path interpolates the first and last shape but not the shapes in between. As a result the motion of the hand tends to this intermediate shapes but never reaches them which enables the generation of arbitrary smooth discrete paths. For Bézier curves the number of input shapes determines the polynomial degree of the underlying curve. In contrast, for discrete B-splines we can fix the underlying polynomial degree, i.e. the applied discrete de Casteljau algorithm, for any number of input shapes by computing additional Bézier control shapes. In Figure 5.15 and 5.16 we depict the extension of discrete quadratic and cubic Bézier curves to discrete quadratic and cubic B-splines in shape space. Unfortunately, the discrete geodesic B-splines do not interpolate the input shapes as their Euclidean counterparts. Therefore, we extended cardinal splines from Euclidean space to Riemannian shape space. In Figure 5.17 and 5.18 we show two discrete cardinal splines in shape space for the same input poses of a hand but for different tension parameter  $\kappa$ . To compute the discrete path, we first computed the Bézier control shapes, whereby the green shapes were computed via discrete geodesics and the pink shapes via discrete parallel transport. Then, we compute the discrete cardinal spline via the corresponding discrete Bézier curve for the four segments. Finally, we present the difference of a discrete geodesic and discrete cardinal splines in dependence of the tension parameter  $\kappa$  in more detail in Figure 5.19 for three input poses of a cactus.

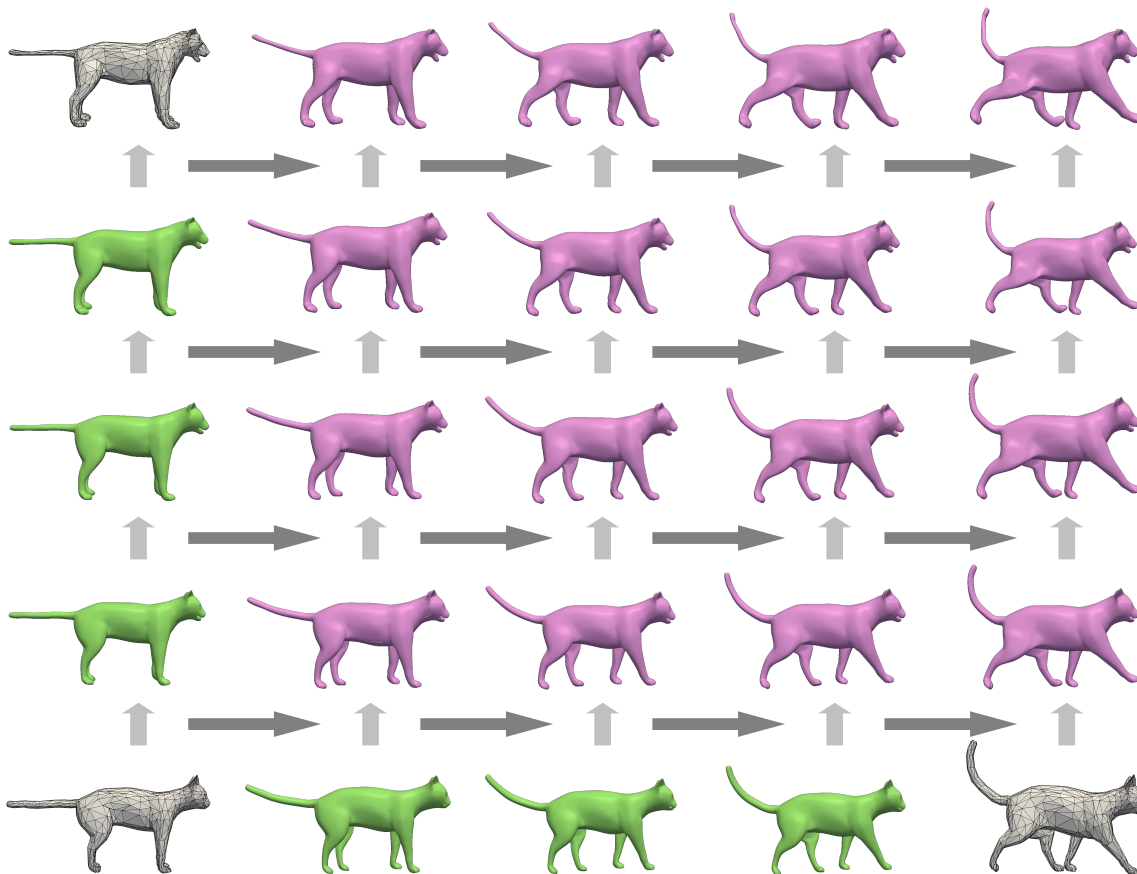


Figure 5.12.: **Discrete Parallel Transport:** Transfer of the motion of a cat as discrete geodesic with  $K = 4$  (bottom) to a lion (top row). The parallel transport was performed in five steps. First, we computed a discrete geodesic with  $K = 4$  between the two similar looking poses of the cat and the lion (first column), then we transferred the motion of the cat iteratively to the shapes on this discrete geodesics until we end up with the motion of the the lion. The three input meshes have the same connectivity with  $\mathcal{I}_v = 1009$  and all computations were done with  $\delta = 0.9$ .

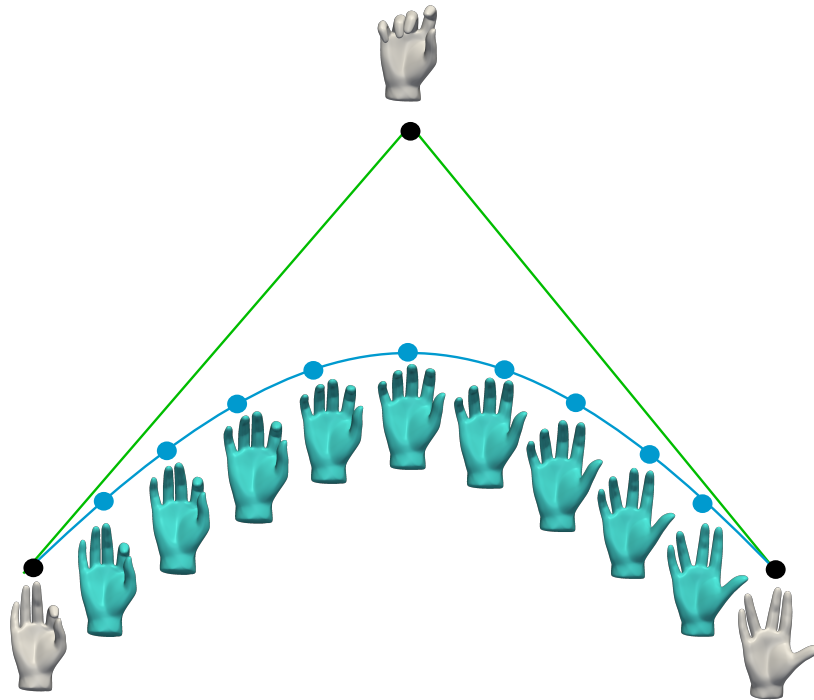


Figure 5.13.: **Discrete Quadratic Bézier Curve:** Visualization of a discrete quadratic Bézier curve  $\mathbf{B}^{10}$  for three input poses of a hand with  $K = 10$  and  $\delta = 0.01$ .

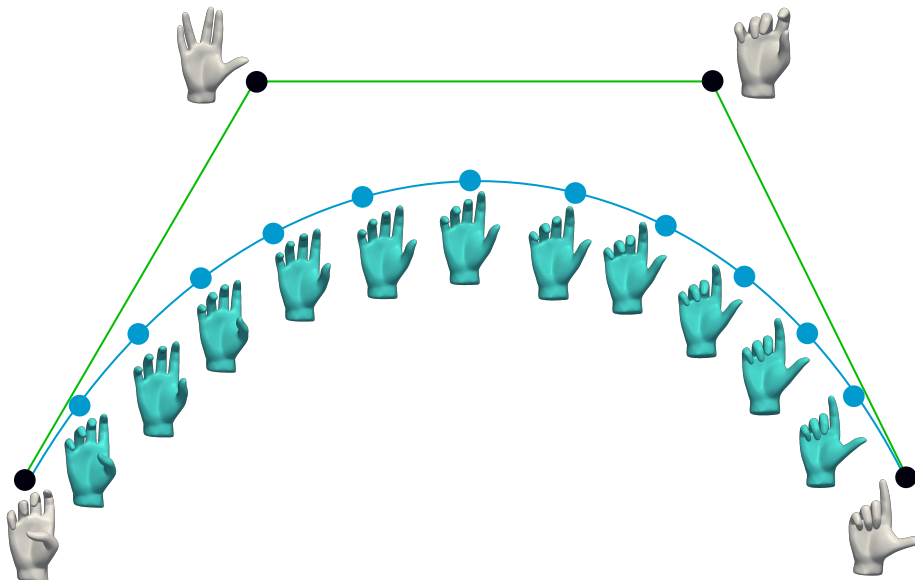


Figure 5.14.: **Discrete Cubic Bézier Curve:** Visualization of a discrete cubic Bézier curve  $\mathbf{B}^{12}$  for four input poses of a hand with  $\delta = 0.01$ .

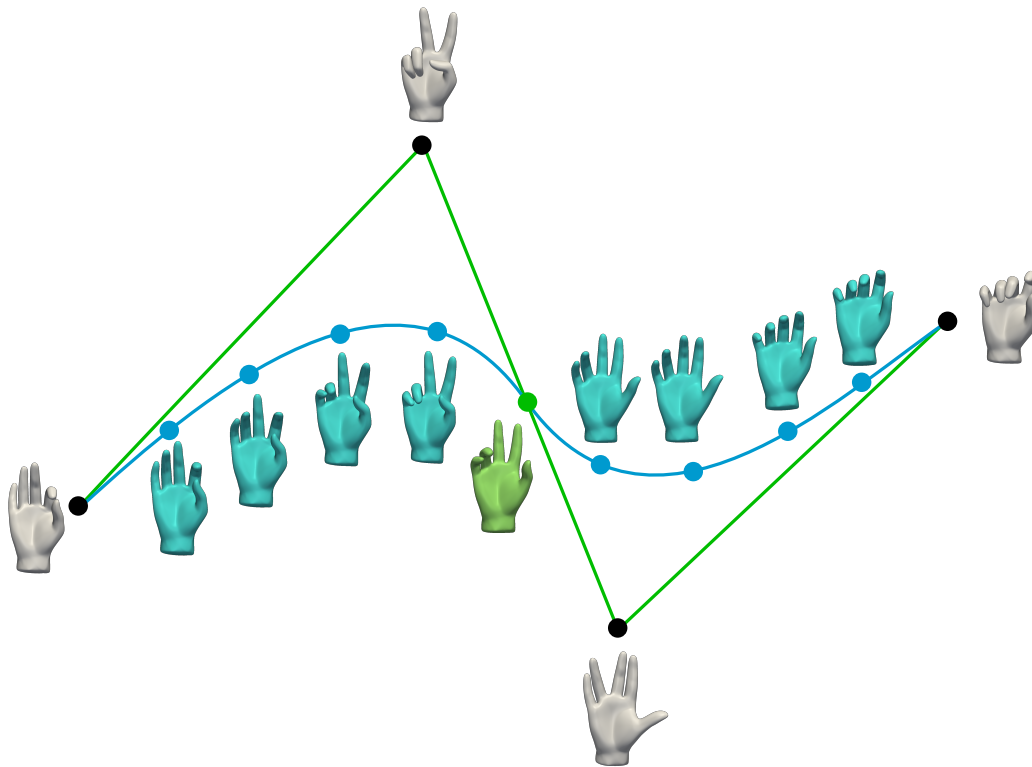


Figure 5.15.: **Discrete Quadratic B-spline:** Visualization of a discrete quadratic B-spline curve  $\text{QBS}^{5,2}$  for four input poses of a hand with  $\delta = 0.01$ . Here, we first computed the missing Bézier control point (green shape) and then performed the discrete de Casteljau algorithm for the two discrete quadratic Bézier curves (cf. with Figure 5.4).

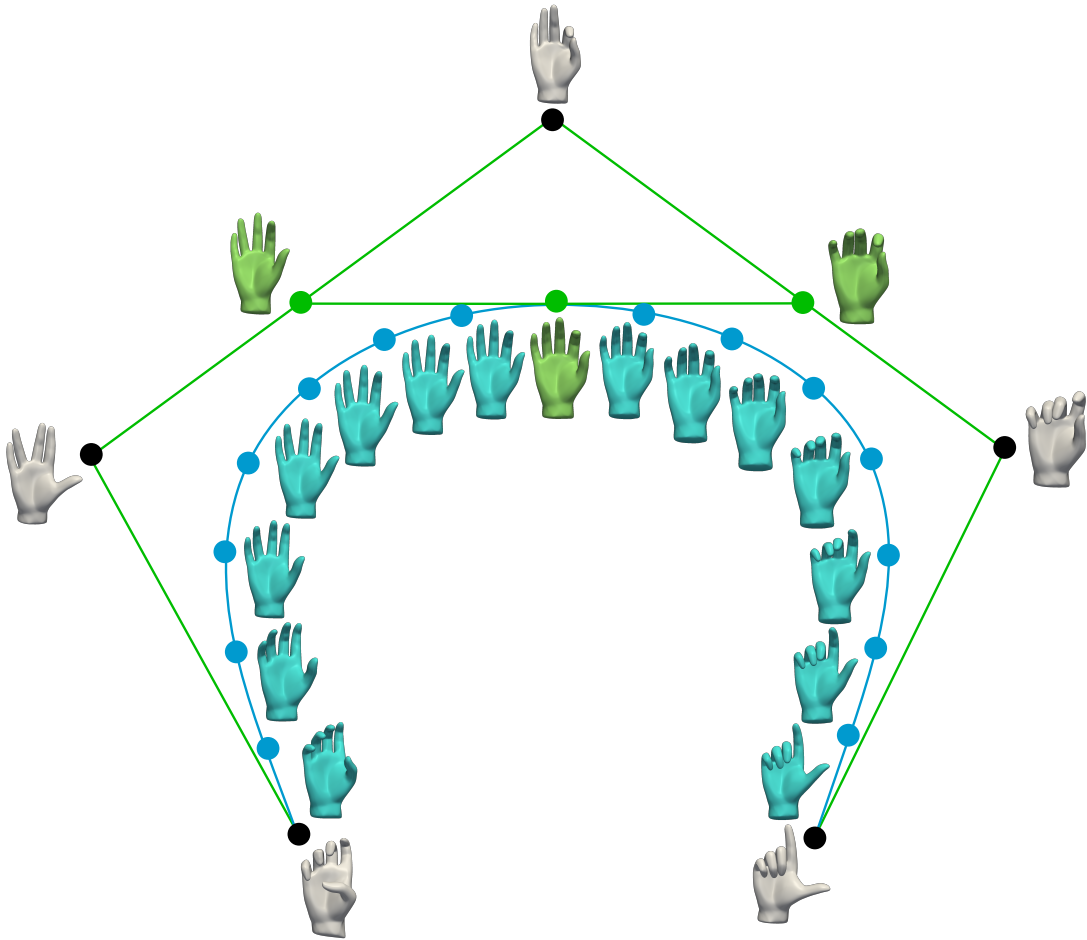


Figure 5.16.: **Discrete Cubic B-spline:** Visualization of a discrete cubic B-spline curve  $\text{CBS}^{8,2}$  for five input poses of a hand with  $\delta = 0.01$ . Here, we first computed the missing three Bézier control points (green shapes) and then performed the discrete de Casteljau algorithm for the two discrete cubic Bézier curves (cf. with Figure 5.5).

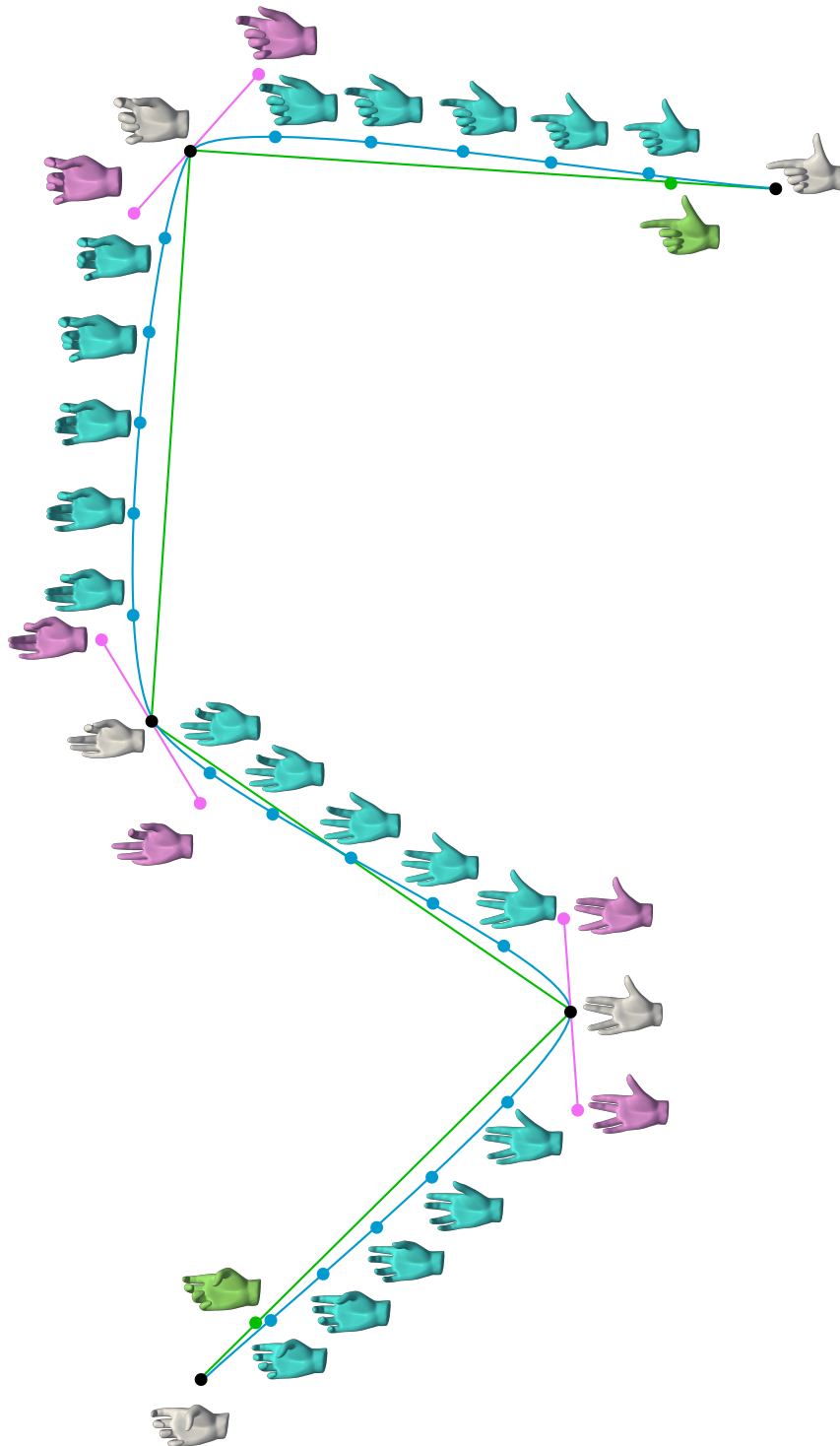


Figure 5.17.: **Discrete Cardinal Spline** ( $\kappa = 1/2$ ): We depict a discrete cardinal spline (gray and cyan shapes) for five input poses of a hand (all gray shapes, the control meshes are depicted in Figure 5.9) as well as the corresponding control shapes of the four discrete cubic Bézier curves (gray, green and pink shapes).

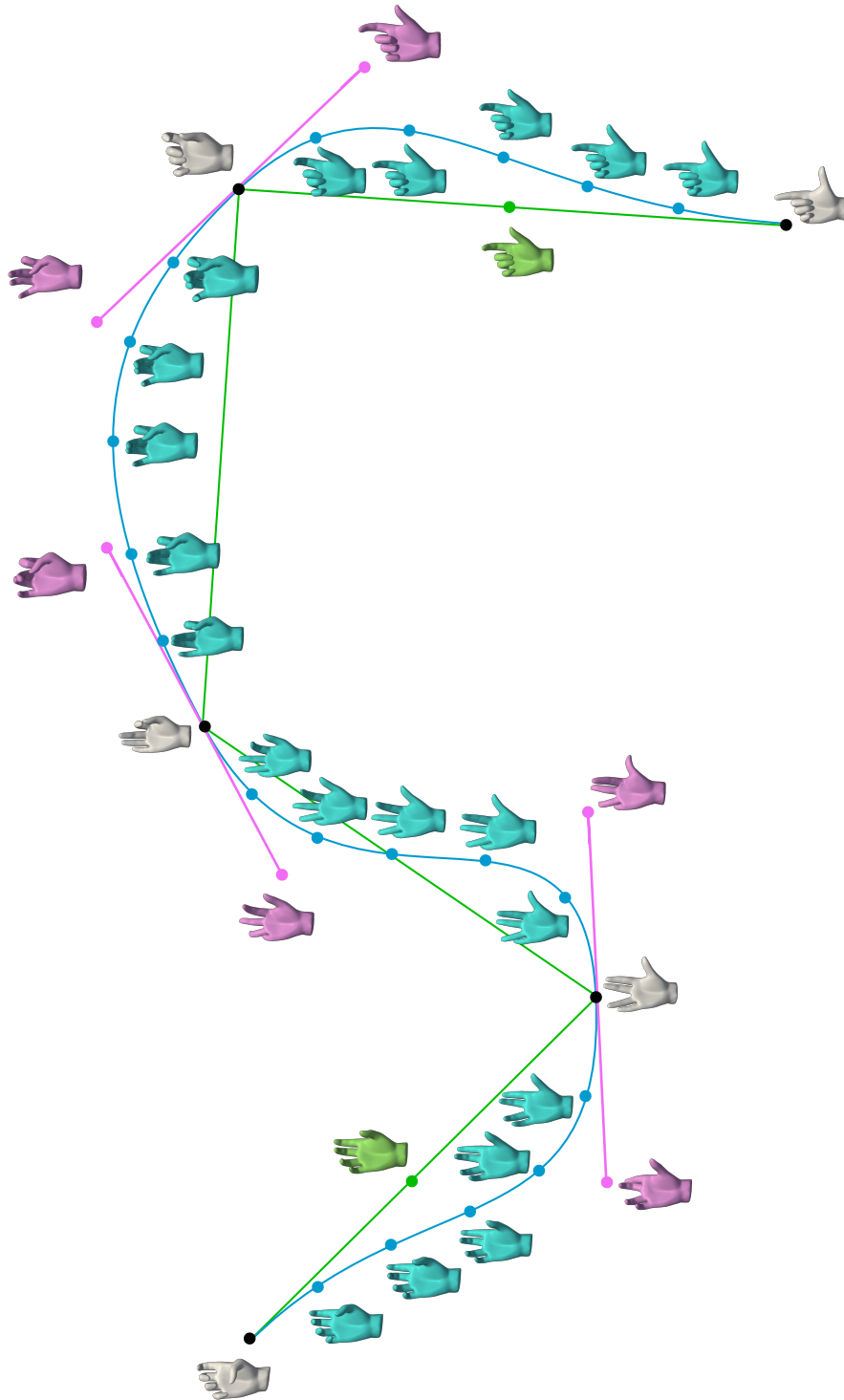


Figure 5.18.: **Discrete Cardinal Spline** ( $\kappa = 3/2$ ): Visualization of a discrete cardinal spline for the same input shapes as in Figure 5.17 but with  $\kappa = 3/2$ .

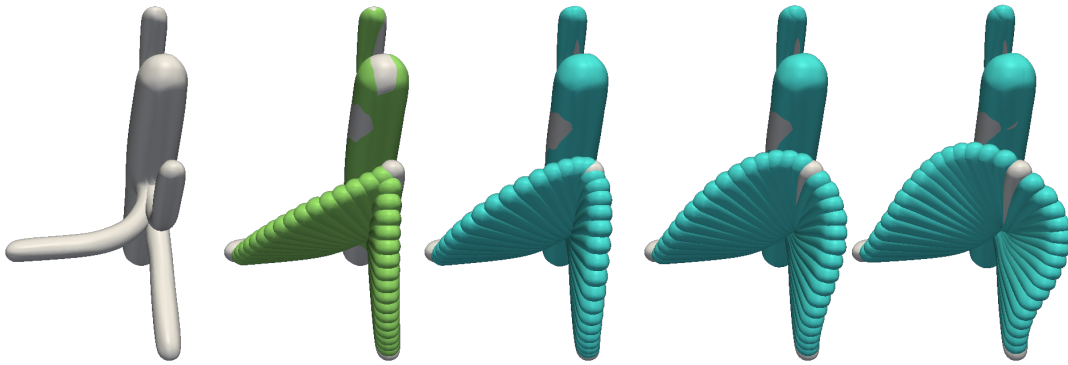


Figure 5.19.: **Influence of  $\kappa$  for Discrete Cardinal Splines:** We depict three input shapes of a cactus (i.e.  $m = 2$ ) on the left, followed by a piecewise discrete geodesic (i.e.  $\kappa = 0$ ) and by three different discrete cardinal splines for  $\kappa = 1/2$ ,  $\kappa = 1$  and  $\kappa = 3/2$ . The input shapes have all the same connectivity with  $|\mathcal{I}_v| = 264$  vertices and all computations were performed with  $K = 17$ . As for the piecewise discrete geodesics, the three discrete cardinal splines still interpolate the input shapes but the paths have more tension for increasing  $\kappa$ .







# A. Appendix

## A.1. Quartic Box Spline Shape Functions

Here we list the 12 shape quartic box spline functions corresponding to the indexing in Fig. 2.5, left.

$$N_1(\xi_1, \xi_2) = \frac{1}{12}(\xi_3^4 + 2\xi_3^3\xi_1)$$

$$N_2(\xi_1, \xi_2) = \frac{1}{12}(\xi_3^4 + 2\xi_3^3\xi_2)$$

$$N_3(\xi_1, \xi_2) = \frac{1}{12}(\xi_3^4 + 2\xi_3^3\xi_2 + 6\xi_3^3\xi_1 + 6\xi_3^2\xi_1\xi_2 + 12\xi_3^2\xi_1^2 + 6\xi_3\xi_1^2\xi_2 + 6\xi_3\xi_1^3 + 2\xi_1^3\xi_2 + \xi_1^4)$$

$$N_4(\xi_1, \xi_2) = \frac{1}{12}(6\xi_3^4 + 24\xi_3^3\xi_2 + 24\xi_3^2\xi_2^2 + 8\xi_3\xi_2^3 + \xi_2^4 + 24\xi_3^3\xi_1 + 60\xi_3^2\xi_1\xi_2 + 36\xi_3\xi_1\xi_2^2 + 6\xi_1\xi_2^3 + 24\xi_3^2\xi_1^2 + 36\xi_3\xi_1^2\xi_2 + 12\xi_1^2\xi_2^2 + 8\xi_3\xi_1^3 + 6\xi_1^3\xi_2 + \xi_1^4)$$

$$N_5(\xi_1, \xi_2) = \frac{1}{12}(\xi_3^4 + 6\xi_3^3\xi_2 + 12\xi_3^2\xi_2^2 + 6\xi_3\xi_2^3 + \xi_2^4 + 2\xi_3^3\xi_1 + 6\xi_3^2\xi_1\xi_2 + 6\xi_3\xi_1\xi_2^2 + 2\xi_1\xi_2^3)$$

$$N_6(\xi_1, \xi_2) = \frac{1}{12}(\xi_1^4 + 2\xi_3\xi_1^3)$$

$$N_7(\xi_1, \xi_2) = \frac{1}{12}(\xi_3^4 + 6\xi_3^3\xi_2 + 12\xi_3^2\xi_2^2 + 6\xi_3\xi_2^3 + \xi_2^4 + 8\xi_3^3\xi_1 + 36\xi_3^2\xi_1\xi_2 + 36\xi_3\xi_1\xi_2^2 + 8\xi_1\xi_2^3 + 24\xi_3^2\xi_1^2 + 60\xi_3\xi_1^2\xi_2 + 24\xi_1^2\xi_2^2 + 24\xi_3\xi_1^3 + 24\xi_1^3\xi_2 + 6\xi_1^4)$$

$$N_8(\xi_1, \xi_2) = \frac{1}{12}(\xi_3^4 + 8\xi_3^3\xi_2 + 24\xi_3^2\xi_2^2 + 24\xi_3\xi_2^3 + 6\xi_2^4 + 6\xi_3^3\xi_1 + 36\xi_3^2\xi_1\xi_2 + 60\xi_3\xi_1\xi_2^2 + 24\xi_1\xi_2^3 + 12\xi_3^2\xi_1^2 + 36\xi_3\xi_1^2\xi_2 + 24\xi_1^2\xi_2^2 + 6\xi_3\xi_1^3 + 8\xi_1^3\xi_2 + \xi_1^4)$$

$$N_9(\xi_1, \xi_2) = \frac{1}{12}(\xi_2^4 + 2\xi_3\xi_2^3)$$

$$N_{10}(\xi_1, \xi_2) = \frac{1}{12}(\xi_1^4 + 2\xi_1^3\xi_2)$$

$$N_{11}(\xi_1, \xi_2) = \frac{1}{12}(2\xi_3\xi_2^3 + \xi_2^4 + 6\xi_3\xi_1\xi_2^2 + 6\xi_1\xi_2^3 + 6\xi_3\xi_1^2\xi_2 + 12\xi_1^2\xi_2^2 + 2\xi_3\xi_1^3 + 6\xi_1^3\xi_2 + \xi_1^4)$$

$$N_{12}(\xi_1, \xi_2) = \frac{1}{12}(\xi_2^4 + 2\xi_2^3\xi_1)$$

where  $\xi_3 = 1 - \xi_1 - \xi_2$ .

## A.2. Additional Subdivision Matrices for Edge Evaluation and Quadrature Points

Here, we list the subdivision matrices for the edge evaluation process if  $N_1, N_2 \neq 4$ .

A. Appendix

---

- $N_1 = 3, N_2 > 4$ :

$$S = \begin{pmatrix} \beta(N_2) & 1 - \beta(N_2) N_2 & \beta(N_2) & \beta(N_2) & 0 & 0 & 0 \\ 1 - \beta(N_1) N_1 & \beta(N_1) & \beta(N_1) & \beta(N_1) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

- $N_1 = 4, N_2 > 4$ :

$$S = \begin{pmatrix} \beta(N_2) & 1 - \beta(N_2) N_2 & \beta(N_2) & 0 & \beta(N_2) & \beta(N_2) & \beta(N_2) & \beta(N_2) \\ 1 - \beta(N_1) N_1 & \beta(N_1) & \beta(N_1) & \beta(N_1) & \beta(N_1) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

- $N_1 = N_2 = 4$ :

$$S = \begin{pmatrix} \beta(N_2) & 1 - \beta(N_2) N_2 & \beta(N_2) & 0 & \beta(N_2) & \beta(N_2) \\ 1 - \beta(N_1) N_1 & \beta(N_1) & \beta(N_1) & \beta(N_1) & \beta(N_1) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

- $N_1 = 3, N_2 = 4$ :

$$S = \begin{pmatrix} \beta(N_2) & 1 - \beta(N_2) N_2 & \beta(N_2) & \beta(N_2) & 0 \\ 1 - \beta(N_1) N_1 & \beta(N_1) & \beta(N_1) & \beta(N_1) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

- $N_1 = 4, N_2 = 3$ :

$$S = \begin{pmatrix} \beta(N_2) & 1 - \beta(N_2) N_2 & \beta(N_2) & 0 & \beta(N_2) \\ 1 - \beta(N_1) N_1 & \beta(N_1) & \beta(N_1) & \beta(N_1) & \beta(N_1) \end{pmatrix}$$

- $N_1 > 4, N_2 = 4$ :

$$S = \begin{pmatrix} \beta(N_2) & 1 - \beta(N_2) N_2 & \beta(N_2) & 0 & 0 & 0 & \beta(N_2) & \beta(N_2) \\ 1 - \beta(N_1) N_1 & \beta(N_1) & \beta(N_1) & \beta(N_1) & \beta(N_1) & \beta(N_1) & \beta(N_1) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} \end{pmatrix}$$

- $N_1 > 4, N_2 = 3$ :

$$S = \begin{pmatrix} \beta(N_2) & 1 - \beta(N_2) N_2 & \beta(N_2) & 0 & 0 & 0 & \beta(N_2) \\ 1 - \beta(N_1) N_1 & \beta(N_1) & \beta(N_1) & \beta(N_1) & \beta(N_1) & \beta(N_1) & \beta(N_1) \end{pmatrix}$$

Furthermore, in Table A.1, A.2, A.3, A.4 and A.5 we list the quadrature points and weights on the unit triangle  $\triangle$  for exact integration of polynomials of degree  $p = 1, p = 2, p = 4, p = 6$  and  $p = 8$ .

$w_q$	$\xi_1^q$	$\xi_2^q$
1.0	$1/3$	$1/3$

Table A.1.: Barycenter quadrature rule: exact for degree  $p = 1$  with  $K = 1$ .

$w_q$	$\xi_1^q$	$\xi_2^q$
$1/6$	$1/2$	0.0
$1/6$	0.0	$1/6$
$1/6$	$1/2$	$1/2$

Table A.2.: Exact Gaussian quadrature rule for degree  $p = 2$  with  $K = 3$ .

$w_q$	$\xi_1^q$	$\xi_2^q$
0.22338158967801	0.44594849091597	0.44594849091597
0.22338158967801	0.44594849091597	0.10810301816807
0.22338158967801	0.10810301816807	0.44594849091597
0.10995174365532	0.09157621350977	0.09157621350977
0.10995174365532	0.09157621350977	0.81684757298046
0.10995174365532	0.81684757298046	0.09157621350977

Table A.3.: Exact Gaussian quadrature rule for degree  $p = 4$  with  $K = 6$ .

$w_q$	$\xi_1^q$	$\xi_2^q$
0.11678627572638	0.24928674517091	0.24928674517091
0.11678627572638	0.24928674517091	0.50142650965818
0.11678627572638	0.50142650965818	0.24928674517091
0.05084490637021	0.06308901449150	0.06308901449150
0.05084490637021	0.06308901449150	0.87382197101700
0.05084490637021	0.87382197101700	0.06308901449150
0.08285107561837	0.31035245103378	0.63650249912140
0.08285107561837	0.63650249912140	0.05314504984482
0.08285107561837	0.05314504984482	0.31035245103378
0.08285107561837	0.63650249912140	0.31035245103378
0.08285107561837	0.31035245103378	0.05314504984482
0.08285107561837	0.05314504984482	0.63650249912140

Table A.4.: Exact Gaussian quadrature rule for degree  $p = 6$  with  $K = 12$ .

$w_q$	$\xi_1^q$	$\xi_2^q$
0.14431560767778	0.3333333333333333	0.3333333333333333
0.09509163426728	0.45929258829272	0.45929258829272
0.09509163426728	0.45929258829272	0.08141482341455
0.09509163426728	0.08141482341455	0.45929258829272
0.10321737053471	0.17056930775176	0.17056930775176
0.10321737053471	0.17056930775176	0.65886138449647
0.10321737053471	0.65886138449647	0.17056930775176
0.03245849762319	0.05054722831703	0.05054722831703
0.03245849762319	0.05054722831703	0.89890554336593
0.03245849762319	0.89890554336593	0.05054722831703
0.02723031417443	0.26311282963463	0.72849239295540
0.02723031417443	0.72849239295540	0.00839477740995
0.02723031417443	0.00839477740995	0.26311282963463
0.02723031417443	0.72849239295540	0.26311282963463
0.02723031417443	0.26311282963463	0.00839477740995
0.02723031417443	0.00839477740995	0.72849239295540

Table A.5.: Exact Gaussian quadrature rule for degree  $p = 8$  with  $K = 16$ .

### A.3. First and Second Discrete Derivatives for Area and Willmore Functional and the Metrics

In this section we list in detail the energies and the metrics used in the examples as well as their derivatives to set up the full Newton method. In the following we consider  $Y = \sum_{i \in \mathcal{I}_v} \bar{Y}_i \cdot \Phi_i$  with  $\bar{Y}_i = (\bar{Y}_i^1, \bar{Y}_i^2, \bar{Y}_i^3)^T$  and  $\dot{Y} = \sum_{i \in \mathcal{I}_v} \bar{Y}_i \cdot \Phi_i$  with  $\bar{Y}_i = (\bar{Y}_i^1, \bar{Y}_i^2, \bar{Y}_i^3)^T$ . Please note, that in the described setup in Chapter 4 the computation of a variation of a functional and then discretizing is equivalent to discretizing the functional and then computing the discrete derivative, i.e.  $\mathbf{E}'[Y](\Psi_i^j) = \frac{\partial \mathbf{E}[Y]}{\partial \bar{Y}_i^j}$ . First, we rewrite the energy functional as

$$\mathbf{E}^{\alpha_0, \alpha_1}[Y] = \int_{\mathcal{M}[Y]} \alpha_0 + \frac{\alpha_1}{2} \mathbf{h}^2 \, da = \int_{\Omega} \alpha_0 \sqrt{D[Y]} + \frac{\alpha_1}{2} \frac{F[Y]^2}{D[Y]^{\frac{5}{2}}} \, d\xi$$

where  $D[Y] = \det G[Y]$ ,  $F[Y] = (G_{22}[Y]\tilde{B}_{11}[Y] + G_{11}[Y]\tilde{B}_{22}[Y] - 2G_{12}[Y]\tilde{B}_{12}[Y])$ ,  $\tilde{B}_{ij}[Y] = Y_{,ij} \cdot \tilde{N}[Y]$  and  $\tilde{N}[Y] = Y_{,1} \times Y_{,2}$ . The first variation is

$$\frac{\partial \mathbf{E}^{\alpha_0, \alpha_1}[Y]}{\partial \bar{Y}_i^j} = \int_{\Omega} \frac{\alpha_0}{2\sqrt{D[Y]}} \frac{\partial D[Y]}{\partial \bar{Y}_i^j} + \alpha_1 \left( \frac{F[Y]}{(D[Y])^{\frac{5}{2}}} \cdot \frac{\partial F[Y]}{\partial \bar{Y}_i^j} - \frac{5}{4} \frac{F[Y]^2}{D[Y]^{\frac{7}{2}}} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_i^j} \right) \, d\xi$$

and the second variation is

$$\frac{\partial^2 \mathbf{E}[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} = \int_{\Omega} \frac{\alpha_0}{2\sqrt{D[Y]}} \frac{\partial^2 D[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} - \frac{\alpha_0}{4(D[Y])^{\frac{3}{2}}} \frac{\partial D[Y]}{\partial \bar{Y}_l^k} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_i^j}$$

$$\begin{aligned}
& + \frac{\alpha_1}{D[Y]^{\frac{5}{2}}} \frac{\partial F[Y]}{\partial \bar{Y}_l^k} \cdot \frac{\partial F[Y]}{\partial \bar{Y}_i^j} + \frac{\alpha_1 F[Y]}{(D[X])^{\frac{5}{2}}} \frac{\partial^2 F[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} \\
& - \frac{5}{2} \frac{\alpha_1 F[Y]}{D[Y]^{\frac{7}{2}}} \left( \frac{\partial F[Y]}{\partial \bar{Y}_l^k} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_i^j} + \frac{\partial F[Y]}{\partial \bar{Y}_i^j} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_l^k} \right) \\
& - \frac{35}{8} \frac{\alpha_1 F[Y]^2}{D[Y]^{\frac{7}{2}}} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_l^k} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_i^j} - \frac{5}{4} \frac{\alpha_1 F[Y]^2}{D[Y]^{\frac{7}{2}}} \cdot \frac{\partial^2 D[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} d\xi.
\end{aligned}$$

The first derivatives of  $D[Y]$  and  $F[Y]$  are

$$\frac{\partial D[Y]}{\partial \bar{Y}_i^j} = 2 \begin{pmatrix} Y_{j,1} \\ Y_{j,2} \end{pmatrix}^T \begin{pmatrix} G_{22}[Y] & -G_{12}[Y] \\ -G_{12}[Y] & G_{11}[Y] \end{pmatrix} \begin{pmatrix} \Phi_{i,1} \\ \Phi_{i,2} \end{pmatrix}$$

and

$$\begin{aligned}
\frac{\partial F[Y]}{\partial \bar{Y}_i^j} & = 2 \begin{pmatrix} Y_{j,1} \\ Y_{j,2} \end{pmatrix}^T \begin{pmatrix} \tilde{B}_{22}[Y] & -\tilde{B}_{12}[Y] \\ -\tilde{B}_{12}[Y] & \tilde{B}_{11}[Y] \end{pmatrix} \begin{pmatrix} \Phi_{i,1} \\ \Phi_{i,2} \end{pmatrix} + G_{22}[Y] \left( \Phi_{i,11} \cdot \tilde{N}_j[Y] + Y_{1,11} \cdot \frac{\partial \tilde{N}[Y]}{\partial \bar{Y}_i^j} \right) \\
& + G_{11}[Y] \left( \Phi_{i,22} \cdot \tilde{N}_j[Y] + Y_{2,22} \cdot \frac{\partial \tilde{N}[Y]}{\partial \bar{Y}_i^j} \right) - 2G_{12}[Y] \left( \Phi_{i,12} \cdot \tilde{N}_j[Y] + Y_{1,12} \cdot \frac{\partial \tilde{N}[Y]}{\partial \bar{Y}_i^j} \right)
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial \tilde{N}[Y]}{\partial \bar{Y}_i^1} & = \begin{pmatrix} 0 \\ Y_{3,1}\Phi_{i,2} - Y_{3,2}\Phi_{i,1} \\ Y_{2,2}\Phi_{i,1} - Y_{2,1}\Phi_{i,2} \end{pmatrix}, \quad \frac{\partial \tilde{N}[Y]}{\partial \bar{Y}_i^2} = \begin{pmatrix} Y_{3,2}\Phi_{i,1} - Y_{3,1}\Phi_{i,2} \\ 0 \\ Y_{1,1}\Phi_{i,2} - Y_{1,2}\Phi_{i,1} \end{pmatrix} \\
\text{and } \frac{\partial \tilde{N}[Y]}{\partial \bar{Y}_i^3} & = \begin{pmatrix} Y_{2,1}\Phi_{i,2} - Y_{2,2}\Phi_{i,1} \\ Y_{1,2}\Phi_{i,1} - Y_{1,1}\Phi_{i,2} \\ 0 \end{pmatrix}.
\end{aligned}$$

Furthermore, the second derivatives of  $D[Y]$  and  $F[Y]$  are

$$\begin{aligned}
\frac{\partial^2 D[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} & = 2 \begin{pmatrix} \Phi_{l,1} \\ \Phi_{l,2} \end{pmatrix}^T \left[ \begin{pmatrix} -Y_{k,2}Y_{j,2} & Y_{k,1}Y_{j,2} - Y_{k,2}Y_{j,1} \\ -Y_{k,1}Y_{j,2} + Y_{k,2}Y_{j,1} & -Y_{k,1}Y_{j,1} \end{pmatrix} \right. \\
& \left. + \delta_{kj} \begin{pmatrix} G_{22}[Y] & -G_{12}[Y] \\ -G_{12}[Y] & G_{11}[Y] \end{pmatrix} \right] \begin{pmatrix} \Phi_{i,1} \\ \Phi_{i,2} \end{pmatrix}
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial^2 F[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} & = \delta_{kj} 2 \left( \Phi_{l,2}\Phi_{i,2}\tilde{B}_{11}[Y] + \Phi_{l,1}\Phi_{i,1}\tilde{B}_{22}[Y] - (\Phi_{l,1}\Phi_{i,2} + \Phi_{l,2}\Phi_{i,1})\tilde{B}_{12}[Y] \right) \\
& + 2(Y_{j,2}\Phi_{i,2}) \left( \Phi_{l,11} \cdot \tilde{N}_k[Y] + Y_{1,11} \cdot \frac{\partial \tilde{N}[Y]}{\partial \bar{Y}_l^k} \right)
\end{aligned}$$



$$\begin{aligned}
 & + 2(Y_{j,1}\Phi_{i,1}) \left( \Phi_{l,22} \cdot \tilde{N}_k[Y] + Y_{,22} \cdot \frac{\partial \tilde{N}[Y]}{\partial \bar{Y}_l^k} \right) \\
 & - 2(Y_{j,1}\Phi_{i,2} + Y_{j,2}\Phi_{i,1}) \left( \Phi_{l,12} \cdot \tilde{N}_k[Y] + Y_{,12} \cdot \frac{\partial \tilde{N}[Y]}{\partial \bar{Y}_l^k} \right) \\
 & + 2(Y_{k,2}\Phi_{l,2}) \left( \Phi_{i,11} \cdot \tilde{N}_j[Y] + Y_{,11} \cdot \frac{\partial \tilde{N}[Y]}{\partial \bar{Y}_i^j} \right) \\
 & + 2(Y_{k,1}\Phi_{l,1}) \left( \Phi_{i,22} \cdot \tilde{N}_j[Y] + Y_{,22} \cdot \frac{\partial \tilde{N}[Y]}{\partial \bar{Y}_i^j} \right) \\
 & - 2(Y_{k,1}\Phi_{l,2} + Y_{k,2}\Phi_{l,1}) \left( \Phi_{i,12} \cdot \tilde{N}_j[Y] + Y_{,12} \cdot \frac{\partial \tilde{N}[Y]}{\partial \bar{Y}_i^j} \right) \\
 & + G_{22}[Y] \left( \Phi_{i,11} \cdot \frac{\partial \tilde{N}_j[Y]}{\partial \bar{Y}_l^k} + \Phi_{l,11} \cdot \frac{\partial \tilde{N}_k[Y]}{\partial \bar{Y}_i^j} + Y_{,11} \cdot \frac{\partial^2 \tilde{N}[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} \right) \\
 & + G_{11}[Y] \left( \Phi_{i,22} \cdot \frac{\partial \tilde{N}_j[Y]}{\partial \bar{Y}_l^k} + \Phi_{l,22} \cdot \frac{\partial \tilde{N}_k[Y]}{\partial \bar{Y}_i^j} + Y_{,22} \cdot \frac{\partial^2 \tilde{N}[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} \right) \\
 & - 2G_{12}[Y] \left( \Phi_{i,12} \cdot \frac{\partial \tilde{N}_j[Y]}{\partial \bar{Y}_l^k} + \Phi_{l,12} \cdot \frac{\partial \tilde{N}_k[Y]}{\partial \bar{Y}_i^j} + Y_{,12} \cdot \frac{\partial^2 \tilde{N}[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} \right)
 \end{aligned}$$

where

$$\begin{aligned}
 \frac{\partial^2 \tilde{N}[Y]}{\partial \bar{Y}_l^1 \partial \bar{Y}_i^1} &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \frac{\partial^2 \tilde{N}[Y]}{\partial \bar{Y}_l^2 \partial \bar{Y}_i^1} = \begin{pmatrix} 0 \\ 0 \\ \Phi_{l,2}\Phi_{i,1} - \Phi_{l,1}\Phi_{i,2} \end{pmatrix}, \quad \frac{\partial^2 \tilde{N}[Y]}{\partial \bar{Y}_l^3 \partial \bar{Y}_i^1} = \begin{pmatrix} 0 & 0 \\ \Phi_{l,1}\Phi_{i,2} - \Phi_{l,2}\Phi_{i,1} & 0 \end{pmatrix}, \\
 \frac{\partial^2 \tilde{N}[Y]}{\partial \bar{Y}_l^1 \partial \bar{Y}_i^2} &= \begin{pmatrix} 0 \\ 0 \\ \Phi_{l,1}\Phi_{i,2} - \Phi_{l,2}\Phi_{i,1} \end{pmatrix}, \quad \frac{\partial^2 \tilde{N}[Y]}{\partial \bar{Y}_l^2 \partial \bar{Y}_i^2} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \frac{\partial^2 \tilde{N}[Y]}{\partial \bar{Y}_l^3 \partial \bar{Y}_i^2} = \begin{pmatrix} \Phi_{l,2}\Phi_{i,1} - \Phi_{l,1}\Phi_{i,2} \\ 0 \\ 0 \end{pmatrix}, \\
 \frac{\partial^2 \tilde{N}[Y]}{\partial \bar{Y}_l^1 \partial \bar{Y}_i^3} &= \begin{pmatrix} 0 \\ \Phi_{l,2}\Phi_{i,1} - \Phi_{l,1}\Phi_{i,2} \\ 0 \end{pmatrix}, \quad \frac{\partial^2 \tilde{N}[Y]}{\partial \bar{Y}_l^2 \partial \bar{Y}_i^3} = \begin{pmatrix} \Phi_{l,1}\Phi_{i,2} - \Phi_{l,2}\Phi_{i,1} \\ 0 \\ 0 \end{pmatrix} \text{ and } \frac{\partial^2 \tilde{N}[Y]}{\partial \bar{Y}_l^3 \partial \bar{Y}_i^3} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.
 \end{aligned}$$

Now, for a basis function  $\Psi_i^j$  of  $\mathcal{V}_h$  the metrics are

$$\begin{aligned}
 \mathbf{G}_Y^{\beta_0, \beta_1}(\dot{Y}, \Psi_i^j) &= \int_{\mathcal{M}[Y]} \dot{Y}_j \Phi_i + \beta_0 \nabla_{\mathcal{M}} \dot{Y}_j \nabla_{\mathcal{M}} \Phi_i + \beta_1 \Delta_{\mathcal{M}} \dot{Y}_j \Delta_{\mathcal{M}} \Phi_i \, da \\
 &= \int_{\Omega} \left( \dot{Y}_j \Phi_i + \beta_0 (\nabla \dot{Y}_j)^T G^{-1}[Y] \nabla \Phi_i + \beta_1 \Delta_{\mathcal{M}} \dot{Y}_j \Delta_{\mathcal{M}} \Phi_i \right) \sqrt{D[Y]} \, d\xi
 \end{aligned}$$

where  $\Delta_{\mathcal{M}} \Phi_i = \frac{1}{D[Y]} K_1[Y, \Phi_i] - \frac{1}{D[Y]^2} K_2[Y, \Phi_i]$  and

$$\Delta_{\mathcal{M}} \dot{Y}_j \Delta_{\mathcal{M}} \Phi_i \sqrt{D[Y]} = \frac{1}{D[Y]^{\frac{3}{2}}} \left( K_1[Y, \dot{Y}_j] - \frac{K_2[Y, \dot{Y}_j]}{D[Y]} \right) \left( K_1[Y, \Phi_i] - \frac{K_2[Y, \Phi_i]}{D[Y]} \right)$$

with

$$K_1[Y, \Phi_i] = (Y_{,2} \cdot Y_{,12} - Y_{,1} \cdot Y_{,22})\Phi_{i,1} + (Y_{,1} \cdot Y_{,12} - Y_{,2} \cdot Y_{,11})\Phi_{i,2} + G_{22}[Y]\Phi_{i,11} \\ - 2G_{12}[Y]\Phi_{i,12} + G_{11}[Y]\Phi_{i,22}$$

and

$$K_2[Y, \Phi_i] = \left[ G_{22}[Y]^2(Y_{,1} \cdot Y_{,11}) + G_{11}[Y]G_{22}[Y](Y_{,2} \cdot Y_{,12}) - G_{12}[Y]G_{22}[Y](Y_{,2} \cdot Y_{,11} + 2Y_{,1} \cdot Y_{,12}) \right. \\ \left. - G_{12}[Y]G_{11}[Y](Y_{,2} \cdot Y_{,22}) + G_{12}[Y]^2(Y_{,2} \cdot Y_{,12} + Y_{,1} \cdot Y_{,22}) \right] \Phi_{i,1} + \left[ G_{11}[Y]^2(Y_{,2} \cdot Y_{,22}) \right. \\ \left. + G_{11}[Y]G_{22}[Y](Y_{,1} \cdot Y_{,12}) - G_{12}[Y]G_{11}[Y](Y_{,1} \cdot Y_{,22} + 2Y_{,2} \cdot Y_{,12}) \right. \\ \left. - G_{12}[Y]G_{22}[Y](Y_{,1} \cdot Y_{,11}) + G_{12}[Y]^2(Y_{,2} \cdot Y_{,11} + Y_{,1} \cdot Y_{,12}) \right] \Phi_{i,2}.$$

Then

$$\mathbf{G}_{Y,Y}^{\beta_0, \beta_1}(\dot{Y}, \Psi_i^j) = \frac{\partial \mathbf{G}_Y^{\beta_0, \beta_1}(\dot{Y}, \Psi_i^j)}{\partial \bar{Y}_l^k} = \int_{\Omega} \frac{\dot{Y}_j \Phi_i}{2\sqrt{D[Y]}} \frac{\partial D[Y]}{\partial \bar{Y}_l^k} \\ - \frac{\beta_0}{2\sqrt{D[Y]}} \begin{pmatrix} \dot{Y}_{j,1} \\ \dot{Y}_{j,2} \end{pmatrix}^T G^{-1}[Y] \begin{pmatrix} \Phi_{i,1} \\ \Phi_{i,2} \end{pmatrix} \frac{\partial D[Y]}{\partial \bar{Y}_l^k} \\ + \frac{\beta_0}{\sqrt{D[Y]}} \begin{pmatrix} \dot{Y}_{j,1} \\ \dot{Y}_{j,2} \end{pmatrix}^T \begin{pmatrix} 2Y_{k,2}\Phi_{l,2} & -Y_{k,1}\Phi_{l,2} - Y_{k,2}\Phi_{l,1} \\ -Y_{k,1}\Phi_{l,2} - Y_{k,2}\Phi_{l,1} & 2Y_{k,1}\Phi_{l,1} \end{pmatrix} \begin{pmatrix} \Phi_{i,1} \\ \Phi_{i,2} \end{pmatrix} \\ - \frac{3\beta_1}{2D[Y]^{\frac{5}{2}}} \frac{\partial D[Y]}{\partial \bar{Y}_l^k} \left( K_1[Y, \dot{Y}_j] - \frac{K_2[Y, \dot{Y}_j]}{D[Y]} \right) \left( K_1[Y, \Phi_i] - \frac{K_2[Y, \Phi_i]}{D[Y]} \right) \\ + \frac{\beta_1}{D[Y]^{\frac{3}{2}}} \left( \frac{\partial K_1[Y, \dot{Y}_j]}{\partial \bar{Y}_l^k} + \frac{K_2[Y, \dot{Y}_j]}{D[Y]^2} \frac{\partial D[Y]}{\partial \bar{Y}_l^k} - \frac{1}{D[Y]} \frac{\partial K_2[Y, \dot{Y}_j]}{\partial \bar{Y}_l^k} \right) \\ \cdot \left( K_1[Y, \Phi_i] - \frac{K_2[Y, \Phi_i]}{D[Y]} \right) \\ + \frac{\beta_1}{D[Y]^{\frac{3}{2}}} \left( K_1[Y, \dot{Y}_j] - \frac{K_2[Y, \dot{Y}_j]}{D[Y]} \right) \\ \cdot \left( \frac{\partial K_1[Y, \Phi_i]}{\partial \bar{Y}_l^k} + \frac{K_2[Y, \Phi_i]}{D[Y]^2} \frac{\partial D[Y]}{\partial \bar{Y}_l^k} - \frac{1}{D[Y]} \frac{\partial K_2[Y, \Phi_i]}{\partial \bar{Y}_l^k} \right) d\xi$$

with

$$\frac{\partial K_1[Y, \Phi_i]}{\partial \bar{Y}_l^k} = \left( \Phi_{l,2}Y_{k,12} + Y_{k,2}\Phi_{l,12} - \Phi_{l,1}Y_{k,22} - Y_{l,1}\Phi_{k,22} \right) \Phi_{i,1} \\ \left( \Phi_{l,1}Y_{k,12} + Y_{k,1}\Phi_{l,12} - \Phi_{l,2}Y_{k,11} - Y_{k,2}\Phi_{l,11} \right) \Phi_{i,2} \\ + 2Y_{k,2}\Phi_{l,2}\Phi_{i,11} - 2(Y_{k,1}\Phi_{l,2} + Y_{k,2}\Phi_{l,1})\Phi_{i,12} + 2Y_{k,1}\Phi_{l,1}\Phi_{i,22}$$

and

$$\begin{aligned}
\frac{\partial K_2[Y, \Phi_i]}{\partial \bar{Y}_l^k} = & \left[ 4G_{22}[Y](\Phi_{l,2}Y_{k,2})(Y_{,1} \cdot Y_{,11}) + G_{22}[Y]^2(\Phi_{l,1}Y_{k,11} + Y_{k,1}\Phi_{l,11}) \right. \\
& + G_{11}[Y]G_{22}[Y](\Phi_{l,2}Y_{k,12} + Y_{k,2}\Phi_{l,12}) \\
& + (2\Phi_{l,1}Y_{k,1}G_{22}[Y] + 2G_{11}[Y]\Phi_{l,2}Y_{k,2})(Y_{,2} \cdot Y_{,12}) \\
& - (G_{22}[Y](\Phi_{l,1}Y_{k,2} + \Phi_{l,2}Y_{k,1}) \\
& + 2G_{12}[Y]\Phi_{l,2}Y_{k,2})(Y_{,2} \cdot Y_{,11} + 2Y_{,1} \cdot Y_{,12}) \\
& - G_{12}[Y]G_{22}[Y](\Phi_{l,2}Y_{k,11} + Y_{k,2}\Phi_{l,11} + 2\Phi_{l,1}Y_{k,12} + 2Y_{k,1}\Phi_{l,12}) \\
& - (G_{11}[Y](\Phi_{l,1}Y_{k,2} + \Phi_{l,2}Y_{k,1}) + 2G_{12}[Y]\Phi_{l,1}Y_{k,1})(Y_{,2} \cdot Y_{,22}) \\
& - G_{12}[Y]G_{11}[Y](\Phi_{l,2}Y_{k,22} + Y_{k,2}\Phi_{l,22}) \\
& + G_{12}[Y]^2(\Phi_{l,2}Y_{k,12} + \Phi_{l,1}Y_{k,22} + Y_{k,2}\Phi_{l,12} + Y_{k,1}\Phi_{l,22}) \\
& \left. + 2G_{12}[Y](\Phi_{l,1}Y_{k,2} + \Phi_{l,2}Y_{k,1}) \cdot (Y_{,2} \cdot Y_{,12} + Y_{,1} \cdot Y_{,22}) \right] \Phi_{i,1} \\
& + \left[ 4G_{11}[Y]\Phi_{l,1}Y_{k,1}(Y_{,2} \cdot Y_{,22}) + G_{11}[Y]^2(\Phi_{l,2}Y_{k,22} + Y_{k,2}\Phi_{l,22}) \right. \\
& + (2G_{22}[Y]\Phi_{l,1}Y_{k,1} + 2G_{11}[Y]\Phi_{l,2}Y_{k,2})(Y_{,1} \cdot Y_{,12}) \\
& + G_{11}[Y]G_{22}[Y](\Phi_{l,1}Y_{k,12} + Y_{k,1}\Phi_{l,12}) \\
& - (2G_{12}[Y]\Phi_{l,1}Y_{k,1} + G_{11}[Y](\Phi_{l,2}Y_{k,1} + \Phi_{l,1}Y_{k,2}))(Y_{,1} \cdot Y_{,22} + 2Y_{,2} \cdot Y_{,12}) \\
& - G_{12}[Y]G_{11}[Y](\Phi_{l,1}Y_{k,22} + 2\Phi_{l,2}Y_{k,12} + Y_{k,1}\Phi_{l,22} + 2Y_{k,2}\Phi_{l,12}) \\
& - (2G_{12}[Y]\Phi_{l,2}Y_{k,2} + G_{22}[Y](\Phi_{l,2}Y_{k,1} + \Phi_{l,1}Y_{k,2}))(Y_{,1} \cdot Y_{,11}) \\
& - G_{12}[Y]G_{22}[Y](\Phi_{l,1}Y_{k,11} + Y_{k,1}\Phi_{l,11}) \\
& + 2G_{12}[Y](\Phi_{l,2}Y_{k,1} + \Phi_{l,1}Y_{k,2})(Y_{,2} \cdot Y_{,11} + Y_{,1} \cdot Y_{,12}) \\
& \left. + G_{12}[Y]^2(\Phi_{l,2}Y_{k,11} + \Phi_{l,1}Y_{k,12} + Y_{k,2}\Phi_{l,11} + Y_{k,1}\Phi_{l,12}) \right] \Phi_{i,2}.
\end{aligned}$$

Let us now come to the volume constraint  $C[Y] = \frac{1}{3} \int_{\mathcal{M}[Y]} Y \cdot N[Y] da - V_0 = \int_{\Omega} Y \cdot \tilde{N}[Y] d\xi - V_0$ . Here, the first variation is

$$\frac{\partial C[Y]}{\partial \bar{Y}_i^j} = \frac{1}{3} \left( \int_{\Omega} \Phi_i \cdot \tilde{N}_j[Y] + Y \cdot \frac{\partial \tilde{N}[Y]}{\partial \bar{Y}_i^j} d\xi \right)$$

and the second

$$\frac{\partial^2 C[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} = \frac{1}{3} \left( \int_{\Omega} \Phi_i \cdot \frac{\partial \tilde{N}_j[Y]}{\partial \bar{Y}_l^k} + \Phi_l \cdot \frac{\partial \tilde{N}_k[Y]}{\partial \bar{Y}_i^j} + Y \cdot \frac{\partial^2 \tilde{N}[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} d\xi \right).$$

## A.4. First and Second Discrete Derivatives for the Elastic Energy in Shape Space

For the computation of discrete geodesics, discrete exponential map, discrete parallel transport, discrete Bézier curves, discrete B-splines and discrete cardinal splines it is to compute the first and second derivatives of the discrete elastic energy  $\mathbf{W}$  with respect to the control points of the underlying Sub-

division surface. Therefore, we consider now two Subdivision functions  $X = \sum_{i \in \mathcal{I}_v} \bar{X}_i \Phi_i$  and  $Y = \sum_{i \in \mathcal{I}_v} \bar{Y}_i \Phi_i$ ,  $\bar{X}_i, \bar{Y}_i \in \mathbb{R}^3$  for the discrete elastic energy  $\mathbf{W}[X, Y] = \mathbf{W}_{\text{mem}}[X, Y] + \delta^2 \mathbf{W}_{\text{bend}}[X, Y]$ . We will split the computation with respect to  $\mathbf{W}_{\text{mem}}[X, Y]$  and  $\mathbf{W}_{\text{bend}}[X, Y]$ .

## Energy functionals

Before we compute the actual derivatives we rewrite the energy functional to set up a recursive algorithm (actually we make use of the chain rule).

### Membrane energy

$$\begin{aligned} \mathbf{W}_{\text{mem}}[X, Y] &= \int_{\mathcal{M}[X]} W_{\text{mem}}(A[X, Y]) \, da \\ &= \int_{\Omega} \left( \frac{\mu}{2} \text{tr}(A[X, Y]) + \frac{\lambda}{4} \det A[X, Y] \right. \\ &\quad \left. - \frac{2\mu + \lambda}{4} \log \det A[X, Y] - \mu - \frac{\lambda}{4} \right) \sqrt{\det G[X]} \, d\xi \end{aligned}$$

where

$$\begin{aligned} A[X, Y] &= G[X]^{-1} G[Y] = \frac{1}{\det G[X]} \begin{pmatrix} G_{22}[X] & -G_{12}[X] \\ -G_{12}[X] & G_{11}[X] \end{pmatrix} \begin{pmatrix} G_{11}[Y] & G_{12}[Y] \\ G_{12}[Y] & G_{22}[Y] \end{pmatrix} \\ &= \frac{1}{\det G[X]} \begin{pmatrix} G_{22}[X]G_{11}[Y] - G_{12}[X]G_{12}[Y] & G_{22}[X]G_{12}[Y] - G_{12}[X]G_{22}[Y] \\ -G_{12}[X]G_{11}[Y] + G_{11}[X]G_{12}[Y] & -G_{12}[X]G_{12}[Y] + G_{11}[X]G_{22}[Y] \end{pmatrix}. \end{aligned}$$

Using this we can write

$$\begin{aligned} \mathbf{W}_{\text{mem}}[X, Y] &= \int_{\Omega} \left( \frac{\mu}{2} \frac{1}{\det G[X]} (G_{22}[X]G_{11}[Y] + G_{11}[X]G_{22}[Y] - 2G_{12}[X]G_{12}[Y]) \right. \\ &\quad \left. + \frac{\lambda \det G[Y]}{4 \det G[X]} - \frac{2\mu + \lambda}{4} (\log \det G[Y] - \log \det G[X]) \right. \\ &\quad \left. - \mu - \frac{\lambda}{4} \right) \sqrt{\det G[X]} \, d\xi \\ &= \int_{\Omega} \left( \frac{\mu}{2} T[X, Y] + \frac{\lambda D[Y]}{4 D[X]} - \frac{2\mu + \lambda}{4} (\log D[Y] - \log D[X]) \right. \\ &\quad \left. - \mu - \frac{\lambda}{4} \right) \sqrt{D[X]} \, d\xi \end{aligned}$$

where

$$T[X, Y] = \frac{1}{D[X]} (G_{22}[X]G_{11}[Y] + G_{11}[X]G_{22}[Y] - 2G_{12}[X]G_{12}[Y]).$$

### Bending energy

$$\begin{aligned}
 \mathbf{W}_{\text{bend}}[X, Y] &= \int_{\mathcal{M}[X]} W_{\text{bend}}(Q[X, Y]) \, da = \int_{\Omega} \text{tr}((G[X]^{-1}(B[X] - B[Y]))^2) \sqrt{\det G[X]} \, d\xi \\
 &= \int_{\Omega} \text{tr}(Z[X, Y]^2) \sqrt{\det G[X]} \, d\xi \\
 &= \int_{\Omega} \left( \sum_{m,p=1}^2 Z_{mp}[X, Y] \cdot Z_{pm}[X, Y] \right) \sqrt{\det G[X]} \, d\xi \\
 &= \int_{\Omega} (Z_{11}[X, Y]^2 + 2 \cdot Z_{12}[X, Y]Z_{21}[X, Y] + Z_{22}[X, Y]^2) \sqrt{\det G[X]} \, d\xi
 \end{aligned}$$

where

$$\begin{aligned}
 Z[X, Y] &= G[X]^{-1}(B[X] - B[Y]) \\
 &= \frac{1}{\det G[X]} \begin{pmatrix} G_{22}[X] & -G_{12}[X] \\ -G_{12}[X] & G_{11}[X] \end{pmatrix} \begin{pmatrix} B_{11}[X] - B_{11}[Y] & B_{12}[X] - B_{12}[Y] \\ B_{12}[X] - B_{12}[Y] & B_{22}[X] - B_{22}[Y] \end{pmatrix}
 \end{aligned}$$

and

$$\begin{aligned}
 Z_{11}[X, Y] &= \frac{G_{22}[X]}{\det G[X]} (B_{11}[X] - B_{11}[Y]) - \frac{G_{12}[X]}{\det G[X]} (B_{21}[X] - B_{21}[Y]) \\
 &= C^{1111}[X, Y] + C^{1212}[X, Y], \\
 Z_{12}[X, Y] &= \frac{G_{22}[X]}{\det G[X]} (B_{12}[X] - B_{12}[Y]) - \frac{G_{12}[X]}{\det G[X]} (B_{22}[X] - B_{22}[Y]) \\
 &= C^{1121}[X, Y] + C^{1222}[X, Y], \\
 Z_{21}[X, Y] &= -\frac{G_{21}[X]}{\det G[X]} (B_{11}[X] - B_{11}[Y]) + \frac{G_{11}[X]}{\det G[X]} (B_{21}[X] - B_{21}[Y]) \\
 &= C^{2111}[X, Y] + C^{2212}[X, Y], \\
 Z_{22}[X, Y] &= -\frac{G_{21}[X]}{\det G[X]} (B_{12}[X] - B_{12}[Y]) + \frac{G_{11}[X]}{\det G[X]} (B_{22}[X] - B_{22}[Y]) \\
 &= C^{2121}[X, Y] + C^{2222}[X, Y],
 \end{aligned}$$

where

$$C^{mnpq}[X, Y] = \frac{\tilde{G}_{mn}^{-1}[X]}{D[X]} \left( \frac{\tilde{B}_{pq}[X]}{\sqrt{D[X]}} - \frac{\tilde{B}_{pq}[Y]}{\sqrt{D[Y]}} \right)$$

and

$$G_{mn}^{-1}[X] = \frac{\tilde{G}_{mn}^{-1}[X]}{D[X]}.$$

## First derivatives of energy functionals

Now we compute the first derivatives of  $\mathbf{W}_{\text{mem}}$  and  $\mathbf{W}_{\text{bend}}$  with respect to  $X$  and  $Y$ , and in dependence of the first derivatives of  $T[X, Y]$ ,  $D[X]$ ,  $D[Y]$  and  $Z[X, Y]$ .

### Membrane energy

$$\begin{aligned} \frac{\partial \mathbf{W}_{\text{mem}}[X, Y]}{\partial \bar{X}_i^j} &= \int_{\Omega} \left( \frac{\mu}{2} \frac{\partial T[X, Y]}{\partial \bar{X}_i^j} - \frac{\lambda}{4} \frac{D[Y]}{(D[X])^2} \cdot \frac{\partial D[X]}{\partial \bar{X}_i^j} + \frac{2\mu + \lambda}{4} \frac{1}{D[X]} \cdot \frac{\partial D[X]}{\partial \bar{X}_i^j} \right) \sqrt{D[X]} \\ &+ \left( \frac{\mu}{2} T[X, Y] + \frac{\lambda}{4} \frac{D[Y]}{D[X]} - \frac{2\mu + \lambda}{4} (\log D[Y] - \log D[X]) - \mu - \frac{\lambda}{4} \right) \\ &\quad \cdot \frac{1}{2\sqrt{D[X]}} \frac{\partial D[X]}{\partial \bar{X}_i^j} d\xi \end{aligned}$$

$$\frac{\partial \mathbf{W}_{\text{mem}}[X, Y]}{\partial \bar{Y}_i^j} = \int_{\Omega} \left( \frac{\mu}{2} \frac{\partial T[X, Y]}{\partial \bar{Y}_i^j} + \frac{\lambda}{4} \frac{1}{D[X]} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_i^j} - \frac{2\mu + \lambda}{4} \frac{1}{D[Y]} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_i^j} \right) \sqrt{D[X]} d\xi$$

### Bending energy

$$\begin{aligned} \frac{\partial \mathbf{W}_{\text{bend}}[X, Y]}{\partial \bar{X}_i^j} &= \int_{\Omega} \left( \sum_{m,p=1}^2 Z_{mp}[X, Y] \cdot \frac{\partial Z_{pm}[X, Y]}{\partial \bar{X}_i^j} + \frac{\partial Z_{mp}[X, Y]}{\partial \bar{X}_i^j} \cdot Z_{pm}[X, Y] \right) \sqrt{D[X]} d\xi \\ &+ \frac{1}{2} \int_{\Omega} \left( \sum_{m,p=1}^2 Z_{mp}[X, Y] \cdot Z_{pm}[X, Y] \right) \frac{1}{\sqrt{D[X]}} \cdot \frac{\partial D[X]}{\partial \bar{X}_i^j} d\xi \end{aligned}$$

$$\frac{\partial \mathbf{W}_{\text{bend}}[X, Y]}{\partial \bar{Y}_i^j} = \int_{\Omega} \left( \sum_{m,p=1}^2 Z_{mp}[X, Y] \cdot \frac{\partial Z_{pm}[X, Y]}{\partial \bar{Y}_i^j} + \frac{\partial Z_{mp}[X, Y]}{\partial \bar{Y}_i^j} \cdot Z_{pm}[X, Y] \right) \sqrt{D[X]} d\xi$$

## Second derivatives of energy functionals

Now we compute the second derivatives of  $\mathbf{W}_{\text{mem}}$  and  $\mathbf{W}_{\text{bend}}$  with respect to  $X$  and  $Y$ , and in dependence of the first and second derivatives of  $T[X, Y]$ ,  $D[X]$ ,  $D[Y]$  and  $Z[X, Y]$ .

### Membrane energy

$$\begin{aligned} \frac{\partial^2 \mathbf{W}_{\text{mem}}[X, Y]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} &= \int_{\Omega} \left( \frac{\mu}{2} \frac{\partial^2 T[X, Y]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} - \frac{\lambda}{4} \frac{D[Y]}{(D[X])^2} \cdot \frac{\partial^2 D[X]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} + \frac{\lambda}{2} \frac{D[Y]}{(D[X])^3} \cdot \frac{\partial D[X]}{\partial \bar{X}_l^k} \cdot \frac{\partial D[X]}{\partial \bar{X}_i^j} \right. \\ &\quad \left. - \frac{2\mu + \lambda}{4} \frac{1}{(D[X])^2} \cdot \frac{\partial D[X]}{\partial \bar{X}_l^k} \cdot \frac{\partial D[X]}{\partial \bar{X}_i^j} + \frac{2\mu + \lambda}{4} \frac{1}{D[X]} \cdot \frac{\partial^2 D[X]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} \right) \sqrt{D[X]} \end{aligned}$$

$$\begin{aligned}
 & + \left( \frac{\mu}{2} \frac{\partial T[X, Y]}{\partial \bar{X}_i^j} - \frac{\lambda}{4} \frac{D[Y]}{(D[X])^2} \cdot \frac{\partial D[X]}{\partial \bar{X}_i^j} + \frac{2\mu + \lambda}{4} \frac{1}{D[X]} \cdot \frac{\partial D[X]}{\partial \bar{X}_i^j} \right) \\
 & \quad \cdot \frac{1}{2\sqrt{D[X]}} \frac{\partial D[X]}{\partial \bar{X}_l^k} \\
 & + \left( \frac{\mu}{2} \frac{\partial T[X, Y]}{\partial \bar{X}_l^k} - \frac{\lambda}{4} \frac{D[Y]}{(D[X])^2} \cdot \frac{\partial D[X]}{\partial \bar{X}_l^k} + \frac{2\mu + \lambda}{4} \frac{1}{D[X]} \cdot \frac{\partial D[X]}{\partial \bar{X}_l^k} \right) \\
 & \quad \cdot \frac{1}{2\sqrt{D[X]}} \frac{\partial D[X]}{\partial \bar{X}_i^j} \\
 & + \left( \frac{\mu}{2} T[X, Y] + \frac{\lambda}{4} \frac{D[Y]}{D[X]} - \frac{2\mu + \lambda}{4} (\log D[Y] - \log D[X]) - \mu - \frac{\lambda}{4} \right) \\
 & \quad \left( -\frac{1}{4(D[X])^{\frac{3}{2}}} \cdot \frac{\partial D[X]}{\partial \bar{X}_l^k} \cdot \frac{\partial D[X]}{\partial \bar{X}_i^j} + \frac{1}{2\sqrt{D[X]}} \frac{\partial^2 D[X]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} \right) d\xi
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial^2 \mathbf{W}_{\text{mem}}[X, Y]}{\partial \bar{Y}_l^k \partial \bar{X}_i^j} & = \int_{\Omega} \left( \frac{\mu}{2} \frac{\partial^2 T[X, Y]}{\partial \bar{Y}_l^k \partial \bar{X}_i^j} - \frac{\lambda}{4} \frac{1}{(D[X])^2} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_l^k} \cdot \frac{\partial D[X]}{\partial \bar{X}_i^j} \right) \sqrt{D[X]} \\
 & + \left( \frac{\mu}{2} \frac{\partial T[X, Y]}{\partial \bar{Y}_l^k} + \frac{\lambda}{4} \frac{1}{D[X]} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_l^k} - \frac{2\mu + \lambda}{4} \frac{1}{D[Y]} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_l^k} \right) \\
 & \quad \cdot \frac{1}{2\sqrt{D[X]}} \frac{\partial D[X]}{\partial \bar{X}_i^j} d\xi
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial^2 \mathbf{W}_{\text{mem}}[X, Y]}{\partial \bar{X}_l^k \partial \bar{Y}_i^j} & = \int_{\Omega} \left( \frac{\mu}{2} \frac{\partial^2 T[X, Y]}{\partial \bar{X}_l^k \partial \bar{Y}_i^j} - \frac{\lambda}{4} \frac{1}{(D[X])^2} \cdot \frac{\partial D[X]}{\partial \bar{X}_l^k} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_i^j} \right) \sqrt{D[X]} \\
 & + \left( \frac{\mu}{2} \frac{\partial T[X, Y]}{\partial \bar{Y}_i^j} + \frac{\lambda}{4} \frac{1}{D[X]} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_i^j} - \frac{2\mu + \lambda}{4} \frac{1}{D[Y]} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_i^j} \right) \\
 & \quad \cdot \frac{1}{2\sqrt{D[X]}} \frac{\partial D[X]}{\partial \bar{X}_l^k} d\xi
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial^2 \mathbf{W}_{\text{mem}}[X, Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} & = \int_{\Omega} \left( \frac{\mu}{2} \frac{\partial^2 T[X, Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} + \frac{\lambda}{4} \frac{1}{D[X]} \cdot \frac{\partial^2 D[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} \right. \\
 & \quad \left. - \frac{2\mu + \lambda}{4} \left( -\frac{1}{(D[Y])^2} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_l^k} \cdot \frac{\partial D[Y]}{\partial \bar{Y}_i^j} + \frac{1}{D[Y]} \cdot \frac{\partial^2 D[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} \right) \right) \sqrt{D[X]} d\xi
 \end{aligned}$$

### Bending energy

$$\frac{\partial^2 \mathbf{W}_{\text{bend}}[X, Y]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} = \int_{\Omega} \left( \sum_{m,p=1}^2 \frac{\partial Z_{mp}[X, Y]}{\partial \bar{X}_l^k} \cdot \frac{\partial Z_{pm}[X, Y]}{\partial \bar{X}_i^j} + \frac{\partial Z_{mp}[X, Y]}{\partial \bar{X}_i^j} \cdot \frac{\partial Z_{pm}[X, Y]}{\partial \bar{X}_l^k} \right)$$

$$\begin{aligned}
& + Z_{mp}[X, Y] \cdot \frac{\partial^2 Z_{pm}[X, Y]}{\partial \bar{X}_l^k \partial \bar{Y}_i^j} + \frac{\partial^2 Z_{mp}[X, Y]}{\partial \bar{X}_l^k \partial \bar{Y}_i^j} \cdot Z_{pm}[X, Y] \Big) \sqrt{D[X]} \\
& + \frac{1}{2} \left( \sum_{m,p=1}^2 Z_{mp}[X, Y] \cdot \frac{\partial Z_{pm}[X, Y]}{\partial X_i^j} + \frac{\partial Z_{mp}[X, Y]}{\partial \bar{X}_i^j} \cdot Z_{pm}[X, Y] \right) \\
& \quad \cdot \frac{1}{\sqrt{D[X]}} \frac{\partial D[X]}{\partial \bar{X}_l^k} \\
& + \frac{1}{2} \left( \sum_{m,p=1}^2 Z_{mp}[X, Y] \cdot \frac{\partial Z_{pm}[X, Y]}{\partial \bar{X}_l^k} + \frac{\partial Z_{mp}[X, Y]}{\partial \bar{X}_l^k} \cdot Z_{pm}[X, Y] \right) \\
& \quad \cdot \frac{1}{\sqrt{D[X]}} \frac{\partial D[X]}{\partial \bar{X}_i^j} \\
& + \left( \sum_{m,p=1}^2 Z_{mp}[X, Y] \cdot Z_{pm}[X, Y] \right) \\
& \quad \cdot \left( -\frac{1}{4} \frac{1}{(D[X])^{\frac{3}{2}}} \cdot \frac{\partial D[X]}{\partial \bar{X}_l^k} \cdot \frac{\partial D[X]}{\partial \bar{X}_i^j} + \frac{1}{2} \frac{1}{\sqrt{D[X]}} \cdot \frac{\partial^2 D[X]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} \right) d\xi
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 \mathbf{W}_{\text{bend}}[X, Y]}{\partial \bar{Y}_l^k \partial \bar{X}_i^j} &= \int_{\Omega} \left( \sum_{m,p=1}^2 \frac{\partial Z_{mp}[X, Y]}{\partial \bar{Y}_l^k} \cdot \frac{\partial Z_{pm}[X, Y]}{\partial \bar{X}_i^j} + \frac{\partial Z_{mp}[X, Y]}{\partial \bar{X}_i^j} \cdot \frac{\partial Z_{pm}[X, Y]}{\partial \bar{Y}_l^k} \right. \\
& + Z_{mp}[X, Y] \cdot \frac{\partial^2 Z_{pm}[X, Y]}{\partial \bar{Y}_l^k \partial \bar{X}_i^j} + \frac{\partial^2 Z_{mp}[X, Y]}{\partial \bar{Y}_l^k \partial \bar{X}_i^j} \cdot Z_{pm}[X, Y] \Big) \sqrt{D[X]} d\xi \\
& + \frac{1}{2} \int_{\Omega} \left( \sum_{m,p=1}^2 Z_{mp}[X, Y] \cdot \frac{\partial Z_{pm}[X, Y]}{\partial \bar{Y}_i^j} + \frac{\partial Z_{mp}[X, Y]}{\partial \bar{Y}_i^j} \cdot Z_{pm}[X, Y] \right) \\
& \quad \cdot \frac{1}{\sqrt{D[X]}} \cdot \frac{\partial D[X]}{\partial \bar{X}_l^k} d\xi
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 \mathbf{W}_{\text{bend}}[X, Y]}{\partial \bar{X}_l^k \partial \bar{Y}_i^j} &= \int_{\Omega} \left( \sum_{m,p=1}^2 \frac{\partial Z_{mp}[X, Y]}{\partial \bar{X}_l^k} \cdot \frac{\partial Z_{pm}[X, Y]}{\partial \bar{Y}_i^j} + \frac{\partial Z_{mp}[X, Y]}{\partial \bar{Y}_i^j} \cdot \frac{\partial Z_{pm}[X, Y]}{\partial \bar{X}_l^k} \right. \\
& + Z_{mp}[X, Y] \cdot \frac{\partial^2 Z_{pm}[X, Y]}{\partial \bar{X}_l^k \partial \bar{Y}_i^j} + \frac{\partial^2 Z_{mp}[X, Y]}{\partial \bar{X}_l^k \partial \bar{Y}_i^j} \cdot Z_{pm}[X, Y] \Big) \sqrt{D[X]} d\xi \\
& + \frac{1}{2} \int_{\Omega} \left( \sum_{m,p=1}^2 Z_{mp}[X, Y] \cdot \frac{\partial Z_{pm}[X, Y]}{\partial \bar{Y}_l^k} + \frac{\partial Z_{mp}[X, Y]}{\partial \bar{Y}_l^k} \cdot Z_{pm}[X, Y] \right) \\
& \quad \cdot \frac{1}{\sqrt{D[X]}} \cdot \frac{\partial D[X]}{\partial \bar{X}_i^j} d\xi
\end{aligned}$$



$$\begin{aligned} \frac{\partial^2 \mathbf{W}_{\text{bend}}[X, Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} &= \int_{\Omega} \left( \sum_{m,p=1}^2 \frac{\partial Z_{mp}[X, Y]}{\partial \bar{Y}_l^k} \cdot \frac{\partial Z_{pm}[X, Y]}{\partial \bar{Y}_i^j} + \frac{\partial Z_{mp}[X, Y]}{\partial \bar{Y}_i^j} \cdot \frac{\partial Z_{pm}[X, Y]}{\partial \bar{Y}_l^k} \right. \\ &\quad \left. + Z_{mp}[X, Y] \cdot \frac{\partial^2 Z_{pm}[X, Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} + \frac{\partial^2 Z_{mp}[X, Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} \cdot Z_{pm}[X, Y] \right) \sqrt{D[X]} \, d\xi \end{aligned}$$

## First Derivatives of $D[X]$ , $T[X, Y]$ and $Z[X, Y]$

### First Derivatives of $D[X]$

$$\begin{aligned} \frac{\partial D[X]}{\partial \bar{X}_i^j} &= 2(G_{22}(X_{j,1}\Phi_{i,1}) + G_{11}(X_{j,2}\Phi_{i,2}) - G_{12}(X_{j,1}\Phi_{i,2} + X_{j,2}\Phi_{i,1})) \\ &= 2 \begin{pmatrix} X_{j,1} \\ X_{j,2} \end{pmatrix}^T \begin{pmatrix} G_{22} & -G_{12} \\ -G_{12} & G_{11} \end{pmatrix} \begin{pmatrix} \Phi_{i,1} \\ \Phi_{i,2} \end{pmatrix} \end{aligned}$$

### First Derivatives of $T[X, Y]$

$$\begin{aligned} \frac{\partial T[X, Y]}{\partial \bar{X}_i^j} &= -\frac{1}{D[X]^2} \frac{\partial D[X]}{\partial \bar{X}_i^j} (G_{22}[X]G_{11}[Y] + G_{11}[X]G_{22}[Y] - 2G_{12}[X]G_{12}[Y]) \\ &\quad + \frac{1}{D[X]} (2G_{11}[Y](\Phi_{i,2}X_{j,2}) + 2G_{22}[Y](\Phi_{i,1}X_{j,1}) - 2G_{12}[Y](\Phi_{i,1}X_{j,2} + \Phi_{i,2}X_{j,1})) \end{aligned}$$

and

$$\frac{\partial T[X, Y]}{\partial \bar{Y}_i^j} = \frac{1}{D[X]} (2G_{22}[X](\Phi_{i,1}Y_{j,1}) + 2G_{11}[X](\Phi_{i,2}Y_{j,2}) - 2G_{12}[X](\Phi_{i,1}Y_{j,2} + \Phi_{i,2}Y_{j,1})).$$

### First Derivatives of $Z[X, Y]$

Because

$$Z_{mp}[X, Y] = C^{m1p1}[X, Y] + C^{m2p2}[X, Y]$$

and

$$\frac{\partial Z_{mp}[X, Y]}{\partial \bar{X}_i^j} = \frac{\partial C^{m1p1}[X, Y]}{\partial \bar{X}_i^j} + \frac{\partial C^{m2p2}[X, Y]}{\partial \bar{X}_i^j}$$

resp.

$$\frac{\partial Z_{mp}[X, Y]}{\partial \bar{Y}_i^j} = \frac{\partial C^{m1p1}[X, Y]}{\partial \bar{Y}_i^j} + \frac{\partial C^{m2p2}[X, Y]}{\partial \bar{Y}_i^j}$$

the computation of the derivatives reduces to compute the derivatives of

$$C^{mnpq}[X, Y] = \frac{\tilde{G}_{mn}^{-1}[X]}{D[X]} \left( \frac{\tilde{B}_{pq}[X]}{\sqrt{D[X]}} - \frac{\tilde{B}_{pq}[Y]}{\sqrt{D[Y]}} \right).$$

Hence,

$$\begin{aligned} \frac{\partial C^{mnpq}[X, Y]}{\partial \bar{X}_i^j} &= \left( \frac{1}{D[X]} \frac{\partial \tilde{G}_{mn}^{-1}[X]}{\partial \bar{X}_i^j} - \frac{\tilde{G}_{mn}^{-1}[X]}{D[X]^2} \frac{\partial D[X]}{\partial \bar{X}_i^j} \right) \left( \frac{\tilde{B}_{pq}[X]}{\sqrt{D[X]}} - \frac{\tilde{B}_{pq}[Y]}{\sqrt{D[Y]}} \right) \\ &+ \frac{\tilde{G}_{mn}^{-1}[X]}{D[X]} \left( \frac{1}{\sqrt{D[X]}} \frac{\partial \tilde{B}_{pq}[X]}{\partial \bar{X}_i^j} - \frac{\tilde{B}_{pq}[X]}{2D[X]^{\frac{3}{2}}} \frac{\partial D[X]}{\partial \bar{X}_i^j} \right) \end{aligned}$$

and

$$\frac{\partial C^{mnpq}[X, Y]}{\partial \bar{Y}_i^j} = \frac{\tilde{G}_{mn}^{-1}[X]}{D[X]} \left( -\frac{1}{\sqrt{D[Y]}} \frac{\partial \tilde{B}_{pq}[Y]}{\partial \bar{Y}_i^j} + \frac{\tilde{B}_{pq}[Y]}{2D[Y]^{\frac{3}{2}}} \frac{\partial D[Y]}{\partial \bar{Y}_i^j} \right)$$

where

$$\frac{\partial \tilde{G}_{mn}^{-1}[X]}{\partial \bar{X}_i^j} = \begin{cases} 2 \cdot X_{j,2} \Phi_{i,2} & m = n = 0, \\ -(X_{j,1} \Phi_{i,2} + X_{j,2} \Phi_{i,1}) & m = 1, n = 0, \\ -(X_{j,1} \Phi_{i,2} + X_{j,2} \Phi_{i,1}) & m = 0, n = 1, \\ 2 \cdot X_{j,1} \Phi_{i,1} & m = n = 1. \end{cases}$$

and

$$\frac{\partial \tilde{B}_{pq}[X]}{\partial \bar{X}_i^j} = \Phi_{i,pq} \tilde{N}_j[X] + X_{,pq} \cdot \frac{\partial \tilde{N}[X]}{\partial \bar{X}_i^j}$$

where

$$\begin{aligned} \tilde{N}[X] &= \begin{pmatrix} X_{2,1} X_{3,2} - X_{3,1} X_{2,2} \\ X_{3,1} X_{1,2} - X_{1,1} X_{3,2} \\ X_{1,1} X_{2,2} - X_{2,1} X_{1,2} \end{pmatrix} & \frac{\partial \tilde{N}[X]}{\partial \bar{X}_i^1} &= \begin{pmatrix} 0 \\ X_{3,1} \Phi_{i,2} - X_{3,2} \Phi_{i,1} \\ X_{2,2} \Phi_{i,1} - X_{2,1} \Phi_{i,2} \end{pmatrix} \\ \frac{\partial \tilde{N}[X]}{\partial \bar{X}_i^2} &= \begin{pmatrix} X_{3,2} \Phi_{i,1} - X_{3,1} \Phi_{i,2} \\ 0 \\ X_{1,1} \Phi_{i,2} - X_{1,2} \Phi_{i,1} \end{pmatrix} & \frac{\partial \tilde{N}[X]}{\partial \bar{X}_i^3} &= \begin{pmatrix} X_{2,1} \Phi_{i,2} - X_{2,2} \Phi_{i,1} \\ X_{1,2} \Phi_{i,1} - X_{1,1} \Phi_{i,2} \\ 0 \end{pmatrix} \end{aligned}$$

## Second Derivatives of $D[X]$ , $T[X, Y]$ and $Z[X, Y]$

### Second Derivatives of $D[X]$

If  $k \neq j$ :

$$\frac{\partial^2 D[X]}{\partial \bar{X}_k^l \partial \bar{X}_i^j} = 2((X_{k,2} \Phi_{l,2})(X_{j,1} \Phi_{i,1}) + (X_{k,1} \Phi_{l,1})(X_{j,2} \Phi_{i,2}))$$

$$\begin{aligned}
& - (X_{k,2}\Phi_{l,1} + X_{k,1}\Phi_{l,2})(X_{j,1}\Phi_{i,2} + X_{j,2}\Phi_{i,1}) \\
& = 2(X_{k,2}X_{j,1}\Phi_{l,2}\Phi_{i,1} + X_{k,1}X_{j,2}\Phi_{l,1}\Phi_{i,2} - X_{k,2}X_{j,1}\Phi_{l,1}\Phi_{i,2} - X_{k,2}X_{j,2}\Phi_{l,1}\Phi_{i,1} \\
& \quad - X_{k,1}X_{j,1}\Phi_{l,2}\Phi_{i,2} - X_{k,1}X_{j,2}\Phi_{l,2}\Phi_{i,1}) \\
& = 2 \begin{pmatrix} \Phi_{l,1} \\ \Phi_{l,2} \end{pmatrix}^T \begin{pmatrix} -X_{k,2}X_{j,2} & X_{k,1}X_{j,2} - X_{k,2}X_{j,1} \\ -X_{k,1}X_{j,2} + X_{k,2}X_{j,1} & -X_{k,1}X_{j,1} \end{pmatrix} \begin{pmatrix} \Phi_{i,1} \\ \Phi_{i,2} \end{pmatrix}
\end{aligned}$$

If  $k = j$ :

$$\begin{aligned}
\frac{\partial^2 D[X]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} & = 2 \begin{pmatrix} \Phi_{l,1} \\ \Phi_{l,2} \end{pmatrix}^T \begin{pmatrix} -X_{k,2}X_{j,2} & X_{k,1}X_{j,2} - X_{k,2}X_{j,1} \\ -X_{k,1}X_{j,2} + X_{k,2}X_{j,1} & -X_{k,1}X_{j,1} \end{pmatrix} \begin{pmatrix} \Phi_{i,1} \\ \Phi_{i,2} \end{pmatrix} \\
& \quad + 2(G_{22}(\Phi_{l,1}\Phi_{i,1}) + G_{11}(\Phi_{l,2}\Phi_{i,2}) - G_{12}(\Phi_{l,1}\Phi_{i,2} + \Phi_{l,2}\Phi_{i,1})) \\
& = 2 \begin{pmatrix} \Phi_{l,1} \\ \Phi_{l,2} \end{pmatrix}^T \begin{pmatrix} -X_{k,2}X_{j,2} & X_{k,1}X_{j,2} - X_{k,2}X_{j,1} \\ -X_{k,1}X_{j,2} + X_{k,2}X_{j,1} & -X_{k,1}X_{j,1} \end{pmatrix} \begin{pmatrix} \Phi_{i,1} \\ \Phi_{i,2} \end{pmatrix} \\
& \quad + 2 \begin{pmatrix} \Phi_{l,1} \\ \Phi_{l,2} \end{pmatrix}^T \begin{pmatrix} G_{22} & -G_{12} \\ -G_{12} & G_{11} \end{pmatrix} \begin{pmatrix} \Phi_{i,1} \\ \Phi_{i,2} \end{pmatrix}
\end{aligned}$$

### Second Derivatives of $T[X, Y]$

$$\begin{aligned}
\frac{\partial^2 T[X, Y]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} & = -\frac{\delta_{jk}}{D[X]^2} \frac{\partial^2 D[X]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} (G_{22}[X]G_{11}[Y] + G_{11}[X]G_{22}[Y] - 2G_{12}[X]G_{12}[Y]) \\
& \quad + \frac{2}{D[X]^3} \frac{\partial D[X]}{\partial \bar{X}_l^k} \frac{\partial D[X]}{\partial \bar{X}_i^j} (G_{22}[X]G_{11}[Y] + G_{11}[X]G_{22}[Y] - 2G_{12}[X]G_{12}[Y]) \\
& \quad - \frac{1}{D[X]^2} \frac{\partial D[X]}{\partial \bar{X}_i^j} (2G_{11}[Y](\Phi_{l,2}X_{k,2}) + 2G_{22}[Y](\Phi_{l,1}X_{k,1})) \\
& \quad - 2G_{12}[Y](\Phi_{l,1}X_{k,2} + \Phi_{l,2}X_{k,1}) - \frac{1}{D[X]^2} \frac{\partial D[X]}{\partial \bar{X}_l^k} (2G_{11}[Y](\Phi_{i,2}X_{j,2}) \\
& \quad + 2G_{22}[Y](\Phi_{i,1}X_{j,1}) - 2G_{12}[Y](\Phi_{i,1}X_{j,2} + \Phi_{i,2}X_{j,1})) \\
& \quad + \frac{\delta_{jk}}{D[X]} (2G_{11}[Y](\Phi_{i,2}\Phi_{l,2}) + 2G_{22}[Y](\Phi_{i,1}\Phi_{l,1}) - 2G_{12}[Y](\Phi_{i,1}\Phi_{l,2} + \Phi_{i,2}\Phi_{l,1}))
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 T[X, Y]}{\partial \bar{Y}_l^k \partial \bar{X}_i^j} & = -\frac{1}{D[X]^2} \frac{\partial D[X]}{\partial \bar{X}_i^j} (2G_{22}[X](\Phi_{l,1}Y_{k,1}) + 2G_{11}[X](\Phi_{l,2}Y_{k,2}) \\
& \quad - 2G_{12}[X](\Phi_{l,1}Y_{k,2} + \Phi_{l,2}Y_{k,1})) + \frac{1}{D[X]} (4(\Phi_{l,1}Y_{k,1})(\Phi_{i,2}X_{j,2}) \\
& \quad + 4(\Phi_{l,2}Y_{k,2})(\Phi_{i,1}X_{j,1}) - 2(\Phi_{l,1}Y_{k,2} + \Phi_{l,2}Y_{k,1})(\Phi_{i,1}X_{j,2} + \Phi_{i,2}X_{j,1}))
\end{aligned}$$

$$\frac{\partial^2 T[X, Y]}{\partial \bar{X}_l^k \partial \bar{Y}_i^j} = -\frac{1}{D[X]^2} \frac{\partial D[X]}{\partial \bar{X}_l^k} (2G_{22}[X](\Phi_{i,1}Y_{j,1}) + 2G_{11}[X](\Phi_{i,2}Y_{j,2})$$

$$\begin{aligned}
& -2G_{12}[X](\Phi_{i,1}Y_{j,2} + \Phi_{i,2}Y_{j,1}) + \frac{1}{D[X]}(4(\Phi_{l,2}X_{k,2})(\Phi_{i,1}Y_{j,1}) \\
& + 4(\Phi_{l,1}X_{k,1})(\Phi_{i,2}Y_{j,2}) - 2(\Phi_{l,1}X_{k,2} + \Phi_{l,2}X_{k,1})(\Phi_{i,1}Y_{j,2} + \Phi_{i,2}Y_{j,1}))
\end{aligned}$$

$$\frac{\partial^2 T[X, Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} = \frac{\delta_{jk}}{D[X]} (2G_{22}[X](\Phi_{i,1}\Phi_{l,1}) + 2G_{11}[X](\Phi_{i,2}\Phi_{l,2}) - 2G_{12}[X](\Phi_{i,1}\Phi_{l,2} + \Phi_{i,2}\Phi_{l,1})).$$

## Second Derivatives of $Z[X, Y]$

As for the first derivatives of  $Z[X, Y]$  the computation reduces to compute the derivatives of  $C^{mnpq}[X, Y]$ .

$$\begin{aligned}
\frac{\partial^2 C^{mnpq}[X, Y]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} &= \left( \frac{1}{D[X]} \frac{\partial^2 \tilde{G}_{mn}^{-1}[X]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} - \frac{1}{D[X]^2} \frac{\partial \tilde{G}_{mn}^{-1}[X]}{\partial \bar{X}_i^j} \frac{\partial D[X]}{\partial \bar{X}_l^k} - \frac{1}{D[X]^2} \frac{\partial \tilde{G}_{mn}^{-1}[X]}{\partial \bar{X}_l^k} \frac{\partial D[X]}{\partial \bar{X}_i^j} \right. \\
&+ 2 \frac{\tilde{G}_{mn}^{-1}[X]}{D[X]^3} \frac{\partial D[X]}{\partial \bar{X}_l^k} \frac{\partial D[X]}{\partial \bar{X}_i^j} - \frac{\tilde{G}_{mn}^{-1}[X]}{D[X]^2} \frac{\partial^2 D[X]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} \left. \right) \left( \frac{\tilde{B}_{pq}[X]}{\sqrt{D[X]}} - \frac{\tilde{B}_{pq}[Y]}{\sqrt{D[Y]}} \right) \\
&+ \left( \frac{1}{D[X]} \frac{\partial \tilde{G}_{mn}^{-1}[X]}{\partial \bar{X}_i^j} - \frac{\tilde{G}_{mn}^{-1}[X]}{D[X]^2} \frac{\partial D[X]}{\partial \bar{X}_i^j} \right) \left( \frac{1}{\sqrt{D[X]}} \frac{\partial \tilde{B}_{pq}[X]}{\partial \bar{X}_l^k} - \frac{\tilde{B}_{pq}[X]}{2D[X]^{\frac{3}{2}}} \frac{\partial D[X]}{\partial \bar{X}_l^k} \right) \\
&+ \left( \frac{1}{D[X]} \frac{\partial \tilde{G}_{mn}^{-1}[X]}{\partial \bar{X}_l^k} - \frac{\tilde{G}_{mn}^{-1}[X]}{D[X]^2} \frac{\partial D[X]}{\partial \bar{X}_l^k} \right) \left( \frac{1}{\sqrt{D[X]}} \frac{\partial \tilde{B}_{pq}[X]}{\partial \bar{X}_i^j} - \frac{\tilde{B}_{pq}[X]}{2D[X]^{\frac{3}{2}}} \frac{\partial D[X]}{\partial \bar{X}_i^j} \right) \\
&+ \frac{\tilde{G}_{mn}^{-1}[X]}{D[X]} \left( \frac{1}{\sqrt{D[X]}} \frac{\partial^2 \tilde{B}_{pq}[X]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} - \frac{1}{2D[X]^{\frac{3}{2}}} \frac{\partial \tilde{B}_{pq}[X]}{\partial \bar{X}_i^j} \frac{\partial D[X]}{\partial \bar{X}_l^k} \right. \\
&\left. - \frac{1}{2D[X]^{\frac{3}{2}}} \frac{\partial \tilde{B}_{pq}[X]}{\partial \bar{X}_l^k} \frac{\partial D[X]}{\partial \bar{X}_i^j} + \frac{3\tilde{B}_{pq}[X]}{4D[X]^{\frac{5}{2}}} \frac{\partial D[X]}{\partial \bar{X}_i^j} \frac{\partial D[X]}{\partial \bar{X}_l^k} - \frac{\tilde{B}_{pq}[X]}{2D[X]^{\frac{3}{2}}} \frac{\partial^2 D[X]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} \right),
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 C^{mnpq}[X, Y]}{\partial \bar{Y}_l^k \partial \bar{X}_i^j} &= \frac{1}{D[X]} \frac{\partial \tilde{G}_{mn}^{-1}[X]}{\partial \bar{X}_i^j} \left( -\frac{1}{\sqrt{D[Y]}} \frac{\partial \tilde{B}_{pq}[Y]}{\partial \bar{Y}_l^k} + \frac{\tilde{B}_{pq}[Y]}{2D[Y]^{\frac{3}{2}}} \frac{\partial D[Y]}{\partial \bar{Y}_l^k} \right) \\
&- \frac{\tilde{G}_{mn}^{-1}[X]}{D[X]^2} \frac{\partial D[X]}{\partial \bar{X}_i^j} \left( -\frac{1}{\sqrt{D[Y]}} \frac{\partial \tilde{B}_{pq}[Y]}{\partial \bar{Y}_l^k} + \frac{\tilde{B}_{pq}[Y]}{2D[Y]^{\frac{3}{2}}} \frac{\partial D[Y]}{\partial \bar{Y}_l^k} \right),
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 C^{mnpq}[X, Y]}{\partial \bar{X}_l^k \partial \bar{Y}_i^j} &= \frac{1}{D[X]} \frac{\partial \tilde{G}_{mn}^{-1}[X]}{\partial \bar{X}_l^k} \left( -\frac{1}{\sqrt{D[Y]}} \frac{\partial \tilde{B}_{pq}[Y]}{\partial \bar{Y}_i^j} + \frac{\tilde{B}_{pq}[Y]}{2D[Y]^{\frac{3}{2}}} \frac{\partial D[Y]}{\partial \bar{Y}_i^j} \right) \\
&- \frac{\tilde{G}_{mn}^{-1}[X]}{D[X]^2} \frac{\partial D[X]}{\partial \bar{X}_l^k} \left( -\frac{1}{\sqrt{D[Y]}} \frac{\partial \tilde{B}_{pq}[Y]}{\partial \bar{Y}_i^j} + \frac{\tilde{B}_{pq}[Y]}{2D[Y]^{\frac{3}{2}}} \frac{\partial D[Y]}{\partial \bar{Y}_i^j} \right),
\end{aligned}$$

and

$$\frac{\partial^2 C^{mnpq}[X, Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} = \frac{\tilde{G}_{mn}^{-1}[X]}{D[X]} \left( -\frac{1}{\sqrt{D[Y]}} \frac{\partial^2 \tilde{B}_{pq}[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} + \frac{1}{2D[Y]^{\frac{3}{2}}} \frac{\partial \tilde{B}_{pq}[Y]}{\partial \bar{Y}_i^j} \frac{\partial D[Y]}{\partial \bar{Y}_l^k} \right)$$

$$+ \frac{1}{2D[Y]^{\frac{3}{2}}} \frac{\partial \tilde{B}_{pq}[Y]}{\partial \bar{Y}_l^k} \frac{\partial D[Y]}{\partial \bar{Y}_i^j} - \frac{3\tilde{B}_{pq}[Y]}{4D[Y]^{\frac{5}{2}}} \frac{\partial D[Y]}{\partial \bar{Y}_i^j} \frac{\partial D[Y]}{\partial \bar{Y}_l^k} + \frac{\tilde{B}_{pq}[Y]}{2D[Y]^{\frac{3}{2}}} \frac{\partial^2 D[Y]}{\partial \bar{Y}_l^k \partial \bar{Y}_i^j} \Big)$$

where

$$\frac{\partial^2 \tilde{G}_{mn}^{-1}[X]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} = \begin{cases} 0 & j \neq k, \\ 2 \cdot \Phi_{l,2} \Phi_{i,2} & j = k, m = n = 0, \\ -(\Phi_{l,1} \Phi_{i,2} + \Phi_{l,2} \Phi_{i,1}) & j = k, m = 1, n = 0, \\ -(\Phi_{l,1} \Phi_{i,2} + \Phi_{l,2} \Phi_{i,1}) & j = k, m = 0, n = 1, \\ 2 \cdot \Phi_{l,1} \Phi_{i,1} & j = k, m = n = 1. \end{cases}$$

and

$$\frac{\partial^2 \tilde{B}_{pq}[X]}{\partial \bar{X}_l^k \partial \bar{X}_i^j} = \Phi_{i,pq} \frac{\partial \tilde{N}_j[X]}{\partial \bar{X}_l^k} + \Phi_{l,pq} \frac{\partial \tilde{N}_k[X]}{\partial \bar{X}_i^j} + X_{,pq} \cdot \frac{\partial^2 \tilde{N}[X]}{\partial \bar{X}_l^k \partial \bar{X}_i^j}$$

where

$$\begin{aligned} \frac{\partial^2 \tilde{N}[X]}{\partial \bar{X}_l^1 \partial \bar{X}_i^1} &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \frac{\partial^2 \tilde{N}[X]}{\partial \bar{X}_l^2 \partial \bar{X}_i^1} = \begin{pmatrix} 0 \\ 0 \\ \Phi_{l,2} \Phi_{i,1} - \Phi_{l,1} \Phi_{i,2} \end{pmatrix}, \quad \frac{\partial^2 \tilde{N}[X]}{\partial \bar{X}_l^3 \partial \bar{X}_i^1} = \begin{pmatrix} 0 \\ \Phi_{l,1} \Phi_{i,2} - \Phi_{l,2} \Phi_{i,1} \\ 0 \end{pmatrix}, \\ \frac{\partial^2 \tilde{N}[X]}{\partial \bar{X}_l^1 \partial \bar{X}_i^2} &= \begin{pmatrix} 0 \\ 0 \\ \Phi_{l,1} \Phi_{i,2} - \Phi_{l,2} \Phi_{i,1} \end{pmatrix}, \quad \frac{\partial^2 \tilde{N}[X]}{\partial \bar{X}_l^2 \partial \bar{X}_i^2} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \frac{\partial^2 \tilde{N}[X]}{\partial \bar{X}_l^3 \partial \bar{X}_i^2} = \begin{pmatrix} \Phi_{l,2} \Phi_{i,1} - \Phi_{l,1} \Phi_{i,2} \\ 0 \\ 0 \end{pmatrix}, \\ \frac{\partial^2 \tilde{N}[X]}{\partial \bar{X}_l^1 \partial \bar{X}_i^3} &= \begin{pmatrix} 0 \\ \Phi_{l,2} \Phi_{i,1} - \Phi_{l,1} \Phi_{i,2} \\ 0 \end{pmatrix}, \quad \frac{\partial^2 \tilde{N}[X]}{\partial \bar{X}_l^2 \partial \bar{X}_i^3} = \begin{pmatrix} \Phi_{l,1} \Phi_{i,2} - \Phi_{l,2} \Phi_{i,1} \\ 0 \\ 0 \end{pmatrix}, \quad \frac{\partial^2 \tilde{N}[X]}{\partial \bar{X}_l^3 \partial \bar{X}_i^3} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \end{aligned}$$



---

## Acknowledgments

I am especially indebted to Martin Rumpf for the supervision and support of my thesis. Furthermore I would like to thank Bert Jüttler, Dhia Mansour and Angelos Mantzaflaris for the excellent cooperation on projects within this thesis. I had such a luck to get the opportunity to participate in the NFN S117 of the Austrian Science Fund. Here, many thanks go to Bert Jüttler, Otmar Scherzer, Walter Zulehner, Ulrich Langer and their work groups for creating an inspiring and friendly atmosphere at all of our research meetings over the years where I learned so much about isogeometric analysis and the spline zoo. Moreover, I am grateful to Bert Jüttler that I could enjoy the hospitality at the Institute of Applied Geometry in Linz during our project cooperation. Furthermore, I thank Christian Lubich for the possibility to collaborate with Dhia Mansour at the Mathematical Institute of the University of Tübingen.

During the work I always received help from my colleagues at the Institute for Numerical Simulation, University of Bonn. Especially, I would like to thank Behrend Heeren. You were the best roommate I could have hoped for, thank you so much. I am also deeply grateful to Benjamin Berkels, Alexander Effland, Benedict Geihe, Martin Lenz, Stefan Simon and Sascha Tölkes for their assistance with any hardware, software, and programming problems and for our numerous valuable mathematical discussions. I had such a great time with you guys, I already miss you so much.

Thanks goes to the Aim Shape data repository for providing some meshes used in this thesis.

For financial as well as non-material support I thank the Bonn International Graduate School in Mathematics, the Hausdorff Center for Mathematics at Bonn University, the Austrian Science Fund and the German Research Foundation.

Last but not least I would like to thank my wife and my family for their continuous support. This thesis is dedicated to you, Julia.





# Bibliography

- [1] P.-A. Absil, P.-Y. Gouzenbourger, P. Striowski, and B. Wirth. Differentiable piecewise-bezier surfaces on riemannian manifolds. 2015.
- [2] Bart Adams, Maks Ovsjanikov, Michael Wand, Hans-Peter Seidel, and Leonidas J. Guibas. Meshless modeling of deformable shapes and their motion. *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2008.
- [3] P. Antolin, A. Buffa, F. Calabro, M. Martinelli, and G. Sangalli. Efficient matrix computation for tensor-product isogeometric analysis: The use of sum factorization. *Computer Methods in Applied Mechanics and Engineering*, 285:817 – 828, 2015.
- [4] M. Antonelli, C.V. Beccari, G. Casciola, R. Ciarloni, and S. Morigi. Subdivision surfaces integrated in a CAD system. *Computer-Aided Design*, 45:1294–1305, 2013.
- [5] Greg Arden. *Approximation Properties of Subdivision Surfaces*. PhD thesis, University of Washington, 2001.
- [6] Thierry Aubin. *Nonlinear analysis on manifolds : Monge-Ampère equations*. Grundlehren der mathematischen Wissenschaften. Springer, New York, 1982.
- [7] Nadine Balzani and Martin Rumpf. A nested variational time discretization for parametric Willmore flow. *Interfaces and Free Boundaries*, 14(4):431–454, 2012.
- [8] D. Bao, S.-S. Chern, and Z. Shen. *An Introduction to Riemann-Finsler Geometry*. Graduate Texts in Mathematics. Springer-Verlag New York, 2000.
- [9] J. W. Barrett, H. Garcke, and R. Nürnberg. A parametric finite element method for fourth order geometric evolution equations. *J. Comp. Phys.*, 222:441–467, 2007.
- [10] J. W. Barrett, H. Garcke, and R. Nürnberg. Parametric approximation of isotropic and anisotropic elastic flow for closed and open curves. *Num. Math.*, 120:489 – 542, 2012.
- [11] J.W. Barrett, H. Garcke, and R. Nürnberg. Parametric approximation of Willmore flow and related geometric evolution equations. *SIAM Journal on scientific computing*, 31:225–253, 2008.
- [12] Martin Bauer and Martins Bruveris. A new Riemannian setting for surface registration. In Xavier Pennec, Sarang Joshi, and Mads Nielsen, editors, *Proceedings of the Mathematical Foundations in Computational Anatomy workshop*, pages 182–193, 2011.
- [13] Martin Bauer, Philipp Harms, and Peter W. Michor. Sobolev metrics on shape space of surfaces. *J. Geom. Mech.*, 3(4):389–438, 2011.
- [14] M. F. Beg, M.I. Miller, A. Trouvé, and L. Younes. Computational anatomy: Computing metrics on anatomical shapes. In *Proceedings of 2002 IEEE ISBI*, pages 341–344, 2002.

- [15] G. Bellettini, V. Caselles, A. Chambolle, and M. Novaga. Crystalline mean curvature flow of convex sets. *Archive for Rational Mechanics and Analysis*, 179(1):109–152, 2006.
- [16] Miklos Bergou, Max Wardetzky, David Harmon, Denis Zorin, and Eitan Grinspun. A quadratic bending model for inextensible surfaces. *Eurographics Symposium on Geometry Processing*, 2006.
- [17] Benjamin Berkels, Alexander Effland, and Martin Rumpf. Time discrete geodesic paths in the space of images. *SIAM J. Imaging Sci.*, 8 (3):1457–1488, 2015.
- [18] I. A. Bobenko and P. Schröder. Discrete Willmore flow. pages 101–110. ACM Press, 2005.
- [19] Ioana Boier-Martin and Denis Zorin. Differentiable Parameterization of Catmull-Clark Subdivision Surfaces. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, pages 155–164, New York, NY, USA, 2004. ACM.
- [20] Andrea Bonito, Ricardo H. Nochetto, and M. Sebastian Pauletti. Parametric FEM for geometric biomembranes. *Journal of Computational Physics*, 229:3171 – 3188, 2010.
- [21] Dietrich Braess. *Finite Elements*. Cambridge University Press, 3rd edition, 2007.
- [22] S. C. Brenner and L. R. Scott. *The mathematical theory of Finite Element methods*. Springer-Verlag, 2002.
- [23] Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. Topology-invariant similarity of nonrigid shapes. *International Journal of Computer Vision*, 81(3):281–301, 2009.
- [24] M. Camarinha, F. Silva Leite, and P. Crouch. On the geometry of Riemannian cubic polynomials. *Diff. Geom. Appl.*, 15:107–135, 2001.
- [25] Thomas J. Cashman. Beyond Catmull-Clark? A survey of advances in subdivision surface methods. *Computer Graphics Forum*, 31(1):42–61, 2012.
- [26] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10:350–355, 1978.
- [27] A. Chambolle and M. Novaga. Convergence of an algorithm for anisotropic mean curvature motion. *SIAM J. Math. Anal.*, 37:1978–1987, 2006.
- [28] Antonin Chambolle. An algorithm for mean curvature motion. *Interfaces and free Boundaries*, 6:195–218, 2004.
- [29] Guillaume Charpiat, Olivier Faugeras, and Renaud Keriven. Approximations of shape metrics and application to shape warping and empirical shape statistics. *Foundations of Computational Mathematics*, 5(1):1–58, 2005.
- [30] P. G. Ciarlet. *The finite element method for elliptic problems*. Number 40 in Classics in applied mathematics. SIAM, 2002.
- [31] F. Cirak and M. Ortiz. Fully C1-conforming subdivision elements for finite deformation thin-shell analysis. *International Journal for Numerical Methods in Engineering*, 51(7):813–833, 2001.

- 
- [32] F. Cirak, M.J. Scott, E.K. Antonsson, M. Ortiz, and P. Schröder. Integrated modeling, finite-element analysis and engineering design for thin-shell structures using subdivision. *Computer-Aided Design*, 34(2):137–148, 2002.
- [33] Fehmi Cirak, Michael Ortiz, and Peter Schröder. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering*, 47(12):2039–72, 2000.
- [34] U. Clarenz. Enclosure theorems for extremals of elliptic parametric functionals. *Calculus of Variations*, 15:313–324, 2002.
- [35] U. Clarenz. The Wulff-shape minimizes an anisotropic Willmore functional. *Interfaces and Free Boundaries*, 6(3):351–359, 2004.
- [36] U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, and R. Rusu. A finite element method for surface restoration with smooth boundary conditions. *Computer Aided Geometric Design*, 21(5):427–445, 2004.
- [37] J. Austin Cottrell, Thomas J. R. Hughes, and Yuri Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons, Chichester, England, 2009.
- [38] Keenan Crane, Ulrich Pinkall, and Peter Schröder. Robust fairing via conformal curvature flow. *ACM Trans. Graph.*, 32, 2013.
- [39] P. Crouch and F. Silva Leite. The dynamic interpolation problem: On Riemannian manifold, Lie groups and symmetric spaces. *Journal of dynamical and control systems*, 1:177–202, 1995.
- [40] W. Dahmen and C. A. Micchelli. Subdivision algorithms for the generation of box spline surfaces. *Computer Aided Geometric Design*, 1:115 – 129, 1984.
- [41] K. Deckelnick, G. Dziuk, and C. M. Elliott. Computation of geometric partial differential equations and mean curvature flow. *Acta Numerica*, 14:139–232, 2005.
- [42] K. Deckelnick and F. Schieweck. Error analysis for the approximation of axisymmetric willmore flow by c1-elements. *Interfaces and Free Boundaries*, 12(4):551–574, 2010.
- [43] M. Desbrun, A. Hirani, M. Leok, and J. Marsden. Discrete exterior calculus, 2005. arXiv:math.DG/0508341 on arxiv.org.
- [44] Mathieu Desbrun, Eva Kanso, and Yiyang Tong. Discrete differential forms for computational modeling. *Oberwolfach Seminars: Discrete differential geometry*, 38:287–324, 2008.
- [45] Manfredo P. do Carmo. *Differential geometry of curves and surfaces*. Prentice Hall, 1976.
- [46] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356–360, 1978.
- [47] D. A. Dunavant. High degree efficient symmetrical Gaussian quadrature rules for the triangle. *Int. J. Numer. Meth. Engng*, 21:1129–1148, 1985.
- [48] D. Dupuis, U. Grenander, and M.I. Miller. Variational problems on flows of diffeomorphisms for image matching. *Quarterly of Applied Mathematics*, 56:587–600, 1998.

- [49] G. Dziuk. Finite elements for the Beltrami operator on arbitrary surfaces. In S. Hildebrandt and R. Leis, editors, *Partial Differential Equations and Calculus of Variations*, Lecture Notes in Mathematics 1357, pages 142–155. Springer, 1988.
- [50] Gerhard Dziuk. Convergence of a semi-discrete scheme for the curve shortening flow. *Mathematical Models and Methods in Applied Sciences*, 4:589–606, 1994.
- [51] Gerhard Dziuk. Discrete anisotropic curve shortening flow. *Siam J. Numer. Anal.*, 36(6):1808–1830, 1999.
- [52] Gerhard Dziuk. Computational parametric Willmore flow. *Numer. Math.*, 111(1):55–80, 2008.
- [53] I. Eckstein, J.-P. Pons, Y. Tong, C.-C. Kuo, and M. Desbrun. Generalized surface flows for mesh processing. In *Eurographics Symposium on Geometry Processing*, 2007.
- [54] Alexander Effland, Martin Rumpf, Stefan Simon, Kirsten Stahn, and Benedikt Wirth. Bézier curves in the space of images. In *Proceedings Scale Space and Variational Methods in Computer Vision*, Lecture Notes in Computer Science. Springer, 2014. to appear.
- [55] J. Ehlers, F. A. E. Pirani, and A. Schild. The geometry of free fall and light propagation. In *General relativity (papers in honour of J. L. Synge)*, pages 63–84. Clarendon Press, Oxford, 1972.
- [56] Charles M. Elliott and Björn Stinner. Modeling and computation of two phase geometric biomembranes using surface finite elements. *Journal of Computational Physics*, 229:6585 – 6612, 2010.
- [57] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 1998.
- [58] Gerald Farin. *Curves and Surfaces for CAD: A Practical Guide*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition, 2002.
- [59] Feng Feng and William S. Klug. Finite element modeling of lipid bilayer membranes. *J. Comput. Physics*, 220:394–408, 2006.
- [60] P. Fletcher and R. Whitaker. Riemannian metrics on the space of solid shapes. In *MICCAI 2006: Med Image Comput Comput Assist Interv.*, 2006.
- [61] Gero Friesecke, Richard James, Maria Giovanna Mora, and Stefan Müller. Derivation of nonlinear bending theory for shells from three-dimensional nonlinear elasticity by Gamma-convergence. *C. R. Math. Acad. Sci. Paris* 336, 8:697–702, 2003.
- [62] Gero Friesecke, Richard D. James, and Stefan Müller. Rigorous derivation of nonlinear plate theory and geometric rigidity. Technical report, Max-Planck-Institut, Leipzig, 2001.
- [63] Gero Friesecke, Richard D. James, and Stefan Müller. A theorem on geometric rigidity and the derivation of nonlinear plate theory from three dimensional elasticity. *Communications on Pure and Applied Mathematics*, 55(11):1461–1506, 2002.
- [64] Stefan Fröhlich and Mario Botsch. Example-driven deformations based on discrete shells. *Computer Graphics Forum* 30, 8:2246–2257, 2011.

- 
- [65] Matthias Fuchs, Bert Jüttler, Otmar Scherzer, and Huaiping Yang. Shape metrics based on elastic deformations. *J. Math. Imaging Vis.*, 35(1):86–102, 2009.
- [66] R. Giambo and F. Giannoni. An analytical theory for Riemmanian cubic polynomials. *IMA Journal of Mathematical Control and Information*, 19:445–460, 2002.
- [67] P.-Y. Gousenbourger, C. Samir, and P.A. Absil. Piecewise-bezier c1 interpolation on riemannian manifolds with application to 2d shape morphing. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 4086–4091, 2014.
- [68] Seth Green, George Turkiyyah, and Duane Storti. Subdivision–Based Multilevel Methods for Large Scale Engineering Simulation of Thin Shells. In *Proceedings of ACM Solid Modeling*, 2002.
- [69] M. Griebel and D. Oeltz. A sparse grid space-time discretization scheme for parabolic problems. *Computing*, 81(1):1–34, 2007.
- [70] Eitan Grinspun, Fehmi Cirak, Peter Schröder, and Michael Ortiz. Non-Linear Mechanics and Collisions for Subdivision Surfaces. Technical report, Caltech, 1999.
- [71] Eitan Grinspun, Yotam Gingold, Jason Reisman, and Denis Zorin. Computing discrete shape operators on general meshes. *Eurographics*, 25, 2006.
- [72] Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. Discrete shells. In *Eurographics/SIGGRAPH Symposium on Computer Animation*, 2003.
- [73] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I (2Nd Revised. Ed.): Nonstiff Problems*. Springer-Verlag New York, Inc., New York, NY, USA, 1993.
- [74] Ernst Hairer and Christian Lubich. Energy-diminishing integration of gradient systems. *IMA J. Numer. Anal.*, 34:452–461, 2014.
- [75] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration, structure-preserving algorithms for ordinary differential equations*, volume 31 of *Springer Series in Computational Mathematics*. Springer, 2006.
- [76] Ernst Hairer and Gerhard Wanner. *Solving ordinary differential equations. II. Stiff and differential-algebraic problems*. Springer series in computational mathematics. Springer, Heidelberg, New York, 2010.
- [77] B. Heeren, M. Rumpf, M. Wardetzky, and B. Wirth. Time-discrete geodesics in the space of shells. *Computer Graphics Forum*, 31(5):1755–1764, 2012.
- [78] Behrend Heeren. Geodätische im Raum von Schalenformen. diploma thesis, 2011.
- [79] Behrend Heeren, Martin Rumpf, Peter Schröder, Max Wardetzky, and Benedikt Wirth. Exploring the geometry of the space of shells. *Computer Graphics Forum*, 33(5):247–256, 2014.
- [80] Klaus Hildebrandt, Christian Schulz, Christoph von Tycowicz, and Konrad Polthier. Interactive surface modeling using modal analysis. *ACM Transactions on Graphics*, 30(5):119:1–11, 2011.

- [81] J. Hinkle, P. Muralidharan, P.T. Fletcher, and S. Joshi. Polynomial regression on riemannian manifolds. In *ECCV*, 2012.
- [82] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39–41):4135–4195, 2005.
- [83] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199(5–8):301–313, 2010.
- [84] Miao Jin, Wei Zeng, Feng Luo, and Xianfeng Gu. Computing teichmüller shape space. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):504–517, 2009.
- [85] Richard Jordan, David Kinderlehrer, and Felix Otto. The Variational Formulation of the Fokker-Planck Equation. *SIAM Journal on Mathematical Analysis*, 29(1):1–17, 1998.
- [86] Bert Jüttler, Angelos Mantzaflaris, Ricardo Perl, and Martin Rumpf. On isogeometric subdivision methods for pdes on surfaces. submitted.
- [87] Arkady Kheyfets, Warner A. Miller, and Gregory A. Newton. Schild’s ladder parallel transport procedure for an arbitrary connection. *Internat. J. Theoret. Phys.*, 39(12):2891–2898, 2000.
- [88] M. Kilian, N. J. Mitra, and H. Pottmann. Geometric modeling in shape space. In *ACM Transactions on Graphics*, volume 26, pages 1–8, 2007.
- [89] Leif Kobbelt, Katja Daubert, and Hans-Peter Seidel. Ray Tracing of Subdivision Surfaces. In G. Drettakis and N.L. Max, editors, *Rendering Techniques '98*, Eurographics, pages 69–80, Vienna, 1998. Springer.
- [90] Jan Krzysztof Kowalski. Application of Box Splines to the Approximation of Sobolev Spaces. *J. Approx. Theory*, 61:53–73, 1990.
- [91] S. Kurtek, E. Klassen, Z. Ding, and A. Srivastava. A novel Riemannian framework for shape analysis of 3d objects. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [92] S. Kurtek, E. Klassen, J. Gore, Z. Ding, and A. Srivastava. Elastic geodesic paths in shape space of parametrized surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to appear, 2011.
- [93] H. LeDret and A. Raoult. The nonlinear membrane model as a variational limit of nonlinear three-dimensional elasticity. *J. Math. Pures Appl.*, 73:549–578, 1995.
- [94] Nicole Lehmann and Ulrich Reif. Notes on the curvature tensor. *Graphical Models*, 74(6):321–325, 2012.
- [95] A. Lew, M. Marsden, M. Ortiz, and M. West. Variational time integrators. *Int. J. Numer. Meth. Engng*, 60:153–212, 2004.
- [96] Xiuwen Liu, Yonggang Shi, Ivo Dinov, and Washington Mio. A computational model of multi-dimensional shape. *International Journal of Computer Vision*, Online First, 2010.
- [97] Charles Loop. Smooth subdivision surfaces based on triangles. Master’s thesis, 1987.

- 
- [98] S. Luckhaus and Th. Sturzenhecker. Implicit time discretization for the mean curvature flow equation. *Calc. Var.*, 3:253–271, 1995.
- [99] D. Mansour, P. Perl, and M. Rumpf. Higher Order Space and Time Discretizations for Parametric Gradient Flows. in preparation.
- [100] A. Mantzaflaris, B. Jüttler, B.N. Khoromskij, and U. Langer. Matrix Generation in Isogeometric Analysis by Low Rank Tensor Approximation. Technical report, NFN G+S: Technical Report No. 19, 2014.
- [101] A. Mantzaflaris and Bert Jüttler. Integration by Interpolation and Look-up for Galerkin-based Isogeometric Analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:373–400, 2015. Isogeometric Analysis Special Issue.
- [102] Peter W. Michor and David Mumford. Riemannian geometries on spaces of plane curves. *J. Eur. Math. Soc.*, 8:1–48, 2006.
- [103] Oliver Nemitz. *Anisotrope Verfahren in der Bildverarbeitung: Gradientenflüsse, Level-Sets und Narrow Bands*. Dissertation, University of Bonn, 2008.
- [104] Thien Nguyen, Kestutis Karciauskas, and Jörg Peters. A comparative study of several classical, discrete differential and isogeometric methods for solving poisson’s equation on the disk. *Axioms*, 3(2):280–299, 2014.
- [105] L. Noakes, G. Heinzinger, and B. Paden. Cubic splines on curved spaces. *IMA Journal of Mathematical Control and Information*, 6:465–473, 1989.
- [106] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York / Berlin, 1999.
- [107] Nadine Olischläger. *Processing Elastic Surfaces and Related Gradient Flows*. Dissertation, University Bonn, 2010.
- [108] Nadine Olischläger and Martin Rumpf. Two step time discretization of Willmore flow. In Edwin R. Hancock, Ralph R. Martin, and Malcolm A. Sabin, editors, *IMA Conference on the Mathematics of Surfaces*, volume 5654 of *Lecture Notes in Computer Science*, pages 278–292. Springer, 2009.
- [109] Felix Otto. The geometry of dissipative evolution equations: the porous medium equation. *Comm. Partial Differential Equations*, 26(1-2):101–174, 2001.
- [110] Ricardo Perl, Paola Pozzi, and Martin Rumpf. A nested variational time discretization for parametric anisotropic willmore flow. In Michael Griebel, editor, *Singular Phenomena and Scaling in Mathematical Models*, pages 221–241. Springer, 2014.
- [111] Jörg Peters and Ulrich Reif. *Subdivision Surfaces*. Springer Series in Geometry and Computing, 2008.
- [112] Jörg Peters and Xiaobin Wu. On the local linear independence of generalized subdivision functions. *SIAM Journal on Numerical Analysis (SINUM)*, 44(6):2389–2407, 2006.
- [113] Tomasz Popiel and Lyle Noakes. Bezier curves and interpolation in riemannian manifolds. *Journal of Approximation Theory*, 148(2):111 – 127, 2007.

- [114] Helmut Pottmann and Michael Hofer. A variational approach to spline curves on surfaces. *Computer Aided Geometric Design*, 22(7):693 – 709, 2005. Geometric Modelling and Differential Geometry.
- [115] Paola Pozzi. On the gradient flow for the anisotropic area functional. *Math. Nachr.*, 285:707 – 726, 2012.
- [116] Ulrich Reif. A unified approach to subdivision algorithms near extraordinary vertices. *Computer Aided Geometric Design*, 12:153–174, 1995.
- [117] Ulrich Reif and Peter Schröder. Curvature integrability of subdivision surfaces. *Advances in Computational Mathematics*, 14(2):157–174, 2001.
- [118] Martin Rumpf and Benedikt Wirth. An elasticity approach to principal modes of shape variation. In *Second International Conference on Scale Space Methods and Variational Methods in Computer Vision*, volume 5567 of *Lecture Notes in Computer Science*, pages 709–720, 2009.
- [119] Martin Rumpf and Benedikt Wirth. A nonlinear elastic shape averaging approach. *SIAM Journal on Imaging Sciences*, 2(3):800–833, 2009.
- [120] Martin Rumpf and Benedikt Wirth. An elasticity-based covariance analysis of shapes. *International Journal of Computer Vision*, 92(3):281–295, 2011.
- [121] Martin Rumpf and Benedikt Wirth. Variational methods in shape analysis. In Otmar Scherzer, editor, *Handbook of Mathematical Methods in Imaging*, pages 1363–1401. Springer, 2011.
- [122] Martin Rumpf and Benedikt Wirth. Discrete geodesic calculus in the space of viscous fluidic objects. *SIAM J. Imaging Sci.*, 6 (4):2581–2602, 2013.
- [123] Martin Rumpf and Benedikt Wirth. Variational time discretization of geodesic calculus. *IMA Journal of Numerical Analysis*, 2014. online first, doi:10.1093/imanum/dru027.
- [124] R. Rusu. An algorithm for the elastic flow of surfaces. *Interfaces and Free Boundaries*, 7:229–239, 2005.
- [125] Robert Schaback and Helmut Werner. *Numerische Mathematik*. Springer, 4th edition, 1992.
- [126] D. Schillinger, S.J. Hossain, and T.J.R. Hughes. Reduced Bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 277:1–45, 2014.
- [127] F. R. Schmidt, M. Clausen, and D. Cremers. Shape matching by variational computation of geodesics on a manifold. In *Pattern Recognition*, volume 4174 of *LNCS*, pages 142–151. Springer, 2006.
- [128] Peter Schröder and Denis Zorin. Subdivision for modeling and animation. SIGGRAPH 2000 Course Notes, 2000.
- [129] Anuj Srivastava, Aastha Jain, Shantanu Joshi, and David Kaziska. Statistical shape models using elastic-string representations. In P.J. Narayanan, editor, *Asian Conference on Computer Vision*, volume 3851 of *LNCS*, pages 612–621, 2006.



- 
- [130] Anuj Srivastava, Eric Klassen, Shantanu H. Joshi, and Ian H. Jermyn. Shape analysis of elastic curves in euclidean spaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(7):1415–1428, july 2011.
- [131] Jos Stam. Evaluation of Loop Subdivision Surfaces. *Computer Graphics Proceedings (SIGGRAPH 1999)*, 1999.
- [132] G. Strang and G. J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, Englewood Cliffs, N.J., 1973.
- [133] G. Sundaramoorthi, A. Yezzi, and A. Mennucci. Sobolev active contours. *International Journal of Computer Vision.*, 73(3):345–366, 2007.
- [134] Ganesh Sundaramoorthi, Andrea Mennucci, Stefano Soatto, and Anthony Yezzi. A new geometric metric in the space of curves, and applications to tracking deforming objects by prediction and filtering. *SIAM Journal on Imaging Sciences*, 4(1):109–145, 2011.
- [135] Alain Trouvé and Francois-Xavier Vialard. Shape splines and stochastic shape evolutions : A second order point of view. *Quartely of Applied Mathematics*, 70:219–251, 2012.
- [136] Christoph von Tycowicz, Christian Schulz, Hans-Peter Seidel, and Klaus Hildebrandt. An efficient construction of reduced deformable objects. *ACM Transactions on Graphics*, 32(6), 2013.
- [137] Georg Umlauf. Analyzing the characteristic map of triangular subdivision schemes. *Constructive Approximation*, 16:145–155, 2000.
- [138] Max Wardetzky, Miklos Bergou, David Harmon, Denis Zorin, and Eitan Grinspun. Discrete quadratic curvature energies. *Computer Aided Geometric Design (CAGD)*, 24:499–518, 2007.
- [139] Tim Winkler, Jens Drieseberg, Marc Alexa, and Kai Hormann. Multi-scale geometry interpolation. *Computer Graphics Forum*, 29(2):309 – 318, 2010. Proceedings of Eurographics.
- [140] B. Wirth, L. Bar, M. Rumpf, and G. Sapiro. Geodesics in shape space via variational time discretization. In *7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 5681 of *Lecture Notes in Computer Science*, pages 288–302, 2009.
- [141] Benedikt Wirth. *Variational methods in shape space*. Dissertation, University Bonn, 2009.
- [142] Benedikt Wirth, Leah Bar, Martin Rumpf, and Guillermo Sapiro. A continuum mechanical approach to geodesics in shape space. *International Journal of Computer Vision*, 93(3):293–318, 2011.
- [143] Chao Zhang, Behrend Heeren, Martin Rumpf, and William Smith. Shell pca: statistical shape modelling in shell space. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [144] U. Zore, B. Jüttler, and J. Kosinka. On the Linear Independence of (Truncated) Hierarchical Subdivision Splines. Technical report, NFN G+S: Technical Report No. 17, 2014.