Data Dependent Energy Operators for Subdivision Surfaces

Thesis by

Ilja Friedel

In Partial Fulfillment of the Requirements

for the Degree of

Master of Science

California Institute of Technology

Pasadena, California

2002

(Submitted June 1, 2002)

Acknowledgements

I would like to thank my advisor Dr. Peter Schröder for his patience and for all his feedback and input, Dr. Igor Guskov for helping me overcome technical difficulties in chapter 3 by giving me a brief on tensor notation, and finally Patrick Mullen for writing a C++ plugin for Maya and spending an incredible amount of time testing different ideas and implementations.

Contents

Chapter 1

Introduction

The problem of finding surfaces of minimum energy was first proposed by J. L. Lagrange and later J.C. Borda in the second half of the 18th century [Nit75]. In the middle of the 19th century J. Plateau motivated a physical analogy between soap bubbles and minimum membrane energy surfaces [Nit75]. Starting in the middle of the 20th century the computer was used for engineering applications such as the fairing of ship hull designs [Tip98].

The past decade has seen a lot of activity in the computer graphics community aimed at finding good algorithms that compute surfaces satisfying different fairness criteria. While there is no generally accepted definition of *fairing*, it usually involves the minimization of an energy functional with respect to given constraints. One reason motivating this new research is the growth of parameters of surfaces that today's computers are capable of handling. Fairing is an operation that promises the (semi-) automatic generation of smooth, high quality surfaces. It is unrealistic to expect a human designer to create such a surface, if more than just a few free parameters are involved. Other applications involving energy functionals include surface estimation or physically based simulation [GKS02].

In addition to the order of derivatives involved, energy functionals are classified as either simplified and parametrization dependent (but quite fast) or as based on parametrization independent quantities of the surface (being highly nonlinear and difficult to handle). As an example the equilibrium of a thin, elastic wooden spline used by draughtsmen for centuries is characterized by the smooth curve minimizing

$$\int_0^1 (\kappa(t))^2 \, dt. \tag{1.1}$$

This functional involves second derivatives and is parametrization independent.

Most of the previous work based on smooth surfaces used tensor product B-splines of degrees 3 to 5 to represent the surface (see for instance [MS92] and [GLW96]). The high degree of smoothness of these surfaces and consequently of the involved integrands allowed the relatively efficient numerical evaluation of the energy integrals using high order two-dimensional quadrature rules. In [MS92] central differences where used to approximate the partial derivatives of the exact energy functionals. Central differences are notoriously expensive operations. A lot of effort has been made to avoid computing them [NW99]. Many applications vastly benefit from easily evaluated gradients and Hessians of the energies.

Subdivision surfaces became increasingly popular in the past decade. This was partially due to their ease of modeling surfaces of arbitrary topology. Catmull-Clark subdivision surfaces are a generalization of bicubic B-spline patches and it was of no surprise that research was performed to extend the simplified energy functionals to extraordinary patches [HKD93]. Unlike bicubic B-spline surfaces Catmull-Clark surfaces are only C^1 continuous at certain points. The authors of [HKD93] encountered difficulties and obtained diverging bending energy integrals. Fortunately the reason for this divergence is of technical nature and not due to inherently unbounded bending energies of subdivision surfaces. The parametrization independent curvature integrals where shown to be finite in [RS01]. Consequently we introduce in chapter 3 corrected simplified energy functionals with help of the characteristic map parametrization. This parametrization was first introduced in [Rei95] and has proven itself a versatile tool for the analysis near extraordinary vertices.

The main results of this thesis are the derivation of simplified, but data dependent, energy operators and the exact computation of their partial derivatives in chapters 4 and 5 (both of which can be computed very quickly). The new operators scale according to the first fundamental form of the subdivision surface and are positioned as a compromise between the extremes of the previously used simplified and exact functionals.

Chapter 2

Setting

In this chapter we recall general properties, which are referenced in the following chapters. The notation is to similar the one used in [GLW96]. Einstein summation notation is used in several formulas.

2.1 The Local Surface

The surfaces considered here are parametrized over their control mesh. The *control mesh* is a polyhedral manifold of two-dimensional faces embedded in $\mathbb{R}^3$. We assume that each of these faces can be parametrized over $\Omega \subset \mathbb{R}^2$. This allows us to have a local parametrization of the surface patch $S : \Omega \to \mathbb{R}^3$

$$S(\bar{u}) = S(u_1, u_2) = (S_1(u_1, u_2), S_2(u_1, u_2), S_3(u_1, u_2))^T. \tag{2.1}$$

For most of this work S refers to a single Catmull-Clark subdivision surface patch of valence N. The symbol R is also used to denote a surface, but in general it refers to a simplified version of S.

2.1.1 Fundamental Forms of a Surface

The first fundamental form of the surface $R(\bar{u})$ is a 2×2 matrix $\mathbf{I}_R(\bar{u}) = \big(g_{ij}(\bar{u})\big)$ with

$$g_{ij}(u_1, u_2) =< \partial_i R(u_1, u_2),\ \partial_j R(u_1, u_2) > \tag{2.2}$$

The inverse of $\mathbf{I}_R$ is $\mathbf{I}_R^{-1} = (g^{ij})$.

The second fundamental form is defined as a 2×2 matrix $\mathbf{II}_R = (h_{ij})$ with entries

$$h_{ij} =< \partial_i \partial_j R, N_R > \tag{2.3}$$

where $N_R = \frac{\partial_1 R \times \partial_2 R}{\|\partial_1 R \times \partial_2 R\|}$ is the normal of the surface $R(u_1, u_2)$.

Finally the Christoffel symbols of the surface R at parameter value $\bar{u}$ are defined as

$$\Gamma^k_{ij} = g^{kl} < \partial_i \partial_j R, \partial_l R > . \tag{2.4}$$

We often express the Christoffel symbols as two 2x2 matrices $\Gamma^1 = (\Gamma^1_{ij})$ and $\Gamma^2 = (\Gamma^2_{ij})$.

2.1.2 Derivatives with Respect to a chosen Parametrization

The gradient of a scalar function $h(\bar{u}) : \Omega \to \mathbb{R}$, which is considered defined on the reference surface $R(\bar{u})$, is computed as

$$\mathbf{grad}_R(h) \quad = \quad g^{jk} \partial_k h \partial_j R. \tag{2.5}$$

and in matrix notation $\mathbf{grad}_R(h) = (\partial_1 h, \partial_2 h) \cdot I_R^{-1} \cdot (\partial_1 R, \partial_2 R)^T$. In this document all gradients are considered to be row-vectors.

The Hessian of h with respect to the reference surface R is given as

$$\mathbf{Hess}_R(h) = \begin{pmatrix} g^{1l}(\partial_1 \partial_l h - \partial_i h \Gamma^i_{1l}) & g^{1l}(\partial_2 \partial_l h - \partial_i h \Gamma^i_{2l}) \\ g^{2l}(\partial_1 \partial_l h - \partial_i h \Gamma^i_{1l}) & g^{2l}(\partial_2 \partial_l h - \partial_i h \Gamma^i_{2l}) \end{pmatrix} \tag{2.6}$$

and in matrix notation $\mathbf{Hess}_R(h) = I_R^{-1} \cdot \big(\mathbf{Hess}(h) - (\partial_1 h \Gamma^1 + \partial_2 h \Gamma^2)\big)$.

All matrices in the last equation are symmetric. But the product of two symmetric matrices doesn't need to be symmetric. This means while $\mathbf{Hess}_S(S_i)$ is symmetric[1], it is not valid to assume the symmetry of $\mathsf{Hess}_R(h)$ for arbitrary reference surfaces R and functions h.[2]

2.1.3 Curvatures of the Surface

The derivatives of a parametrized surface are not very good indicators for the behavior of the surface. The main reason is the dependence of the derivatives on the chosen parametrization. Better suited are quantities, that are independent of the parametrization, as the *principal curvatures* κ_1, κ_2, the *mean curvature* $\frac{1}{2}(\kappa_1 + \kappa_2)$ or the *Gaussian curvature* $\kappa_1 \cdot \kappa_2$. The principal curvatures of $S(\bar{u})$ are the eigen values of the matrix $\mathbf{I}_S^{-1} \cdot \mathbf{II}_S$

Our main interest is focused on the functional $\int_\Omega \kappa_1^2 + \kappa_2^2 \, d\omega$. This integrand is given [GLW96]

[1] The *Weingarten map* $\mathbf{I}_S^{-1} \cdot \mathbf{II}_S$ is self-adjoint, see [Tho79], page 58.

[2] Example: $R(u_1, u_2) = (u_1, 2u_2, 0)$ and $h(u_1, u_2) = u_1 \cdot u_2$.

by

$$\sum_{i=1}^{3} \mathbf{trace}\big(\mathbf{Hess}_S(S_i) \cdot \mathbf{Hess}_S(S_i)^T\big) \quad = \quad \kappa_1^2 + \kappa_2^2. \tag{2.7}$$

2.2 Data Dependent Energies

The data dependent *membrane* or *stretching energy* of a surface S with respect to the reference surface R is defined as

$$E_R^{\mathbf{memb}}(S) = \sum_{i=1}^{3} \int_\Omega \mathbf{grad}_R(S_i) \cdot \mathbf{grad}_R(S_i)^T \, d\omega_R. \tag{2.8}$$

The data dependent *bending energy*[3] of a patch S with respect to the induced parametrization of R is defined as

$$E_R^{\mathbf{bend}}(S) = \sum_{i=1}^{3} \int_\Omega \mathbf{trace}(\mathbf{Hess}_R(S_i) \cdot \mathbf{Hess}_R(S_i)^T) \, d\omega_R. \tag{2.9}$$

In case $R(\bar{u}) = S(\bar{u})$ for all $\bar{u} \in \Omega$ it follows from equation 2.7, that

$$E_S^{\mathbf{bend}}(S) = \int_\Omega \kappa_1^2 + \kappa_2^2 \, d\omega_S, \tag{2.10}$$

hence the energy functional $E_S^{\mathbf{bend}}(S)$ is independent of parametrization. It was also argued in [GLW96], that if the fundamental forms of S and R are approximately equal for all parameter values $\bar{u}$, then one can expect to have $E_R^{\mathbf{bend}}(S) \approx E_S^{\mathbf{bend}}(S)$. This observation is a major motivation for the work presented here.

2.2.1 Rotation and Translation Invariance of $E_R(S)$

Observe, that only the derivatives of the reference surface R are used to define $E_R(S)$. Hence $E_R(S)$ independent of translations of R.

Let $\bar{R} = QR$, $Q \in Mat(3,3)$, be a linearly transformed version of the reference surface. It is simple to verify that $\partial_i \bar{R} = Q \cdot \partial_i R$ as well as $\partial_i \partial_j \bar{R} = Q \cdot \partial_i \partial_j R$. Now specifically Q be a rotation matrix, e.g. $Q^T Q = I$. Then $I_{\bar{R}} = (< \partial_i \bar{R}, \partial_j \bar{R} >) = (\partial_i \bar{R}^T \cdot Q^T \cdot Q \cdot \partial_j \bar{R}) = (< \partial_i R, \partial_j R >) = I_R$. With the same reasoning we find the Christoffel symbols of R and $\bar{R}$ to be identical. This means that $Hess_R(h) = Hess_{\bar{R}}(h)$ and finally $E_R(S) = E_{\bar{R}}(S)$, e.g. the data dependent energy is invariant under rotations of the reference surface R.

[3] The definition is slightly changed compared to [GLW96] due to the possible asymmetry of $\mathbf{Hess}_R(h)$. This change is necessary to guarantee the positive semi-definiteness of J_S in theorem 1 part b.

2.3 The Stiffness Matrix K

We assume that it is possible to represent the surface S as a linear combination of finitely many basis functions

$$\mathsf{S}(u_1, u_2) = \sum_{i=1}^{M} P_i \cdot N_i(u_1, u_2) \tag{2.11}$$

where the $P_i \in \mathbb{R}^3$ are the control points. Combining this representation of S with equations 2.8 resp. 2.9 one should observe, that **trace**, **grad** and **Hess** are linear with respect to the control points P_i. This allows rewriting the energies as

$$E_R(S) = P^T \cdot K_R \cdot P = \sum_{i,j=1}^{M} K_{ij} \cdot P_i^T \cdot P_j \tag{2.12}$$

where K_P is an $M \times M$ matrix with entries defined as

$$K_{ij}^{\mathbf{memb}} = \int_{\Omega} \mathbf{trace}(\mathbf{grad}_R(N_i) \cdot \mathbf{grad}_R(N_j)^T)\, d\omega_R, \tag{2.13}$$

$$K_{ij}^{\mathbf{bend}} = \int_{\Omega} \mathbf{trace}(\mathbf{Hess}_R(N_i) \cdot \mathbf{Hess}_R(N_j)^T)\, d\omega_R. \tag{2.14}$$

The concept of stiffness matrices partially separates the process of computing the energies of a given surface from the computation of the integrals in equations 2.8 resp. 2.9. We will also see in chapter 5 that stiffness matrices carry important information over the derivatives of the data dependent energies.

2.4 Catmull-Clark Subdivision Surfaces

In this thesis we deal exclusively with Catmull-Clark subdivision surfaces. Nevertheless the ideas expressed here should carry over to other subdivision surfaces as long as the derivative integrals over regular regions exist. This is for instance the case for Doo-Sabin subdivision surfaces, which in general are only C^1.

It is known from [Sta98] that one can represent a Catmull-Clark subdivision patch with a single extraordinary vertex of valence N (see figure 2.1) as an expansion in eigen basis of dimension $M = 2N + 8$

$$S(u_1, u_2) = \sum_{i=1}^{M} C_i \cdot \phi_i(u_1, u_2). \tag{2.15}$$

The $\phi_i : \mathbb{R} \to \mathbb{R}$ are the *eigen basis functions* and the C_i are the control points P_i projected into eigen space. It was shown also in [Sta98] that it is possible to evaluate ϕ_i and its derivatives exactly.

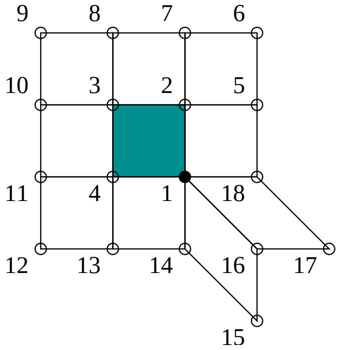


Figure 2.1: Topological neighborhood of an extraordinary patch of valence 5.

A remarkable property of the eigen basis functions is the scaling relation

$$\phi_i(\frac{1}{2}x) = \lambda_i \cdot \phi_i(x). \tag{2.16}$$

The $\lambda_i \in \mathbb{R}$ are the eigen values of the subdivision operator. We assume sorted eigen values $\lambda_i \geq \lambda_{i+1}$. It is known that $\lambda_1 = 1$ and $\lambda := \lambda_2 = \lambda_3 > \lambda_4 =: \mu$ for many interesting subdivision schemes [Rei98].

Chapter 3

Simplified Energies

In [HKD93] the choice for the reference surface R was a generic embedding of the parameter space $[0,1]^2$ into $\mathbb{R}^3$. This simplifies the formula for the membrane energy

$$E^{\mathbf{memb}}_{[0,1]^2}(S) = \int_{[0,1]^2} \|\partial_1 S(\bar{u})\|^2 + \|\partial_2 S(\bar{u})\|^2 \, d\bar{u} \tag{3.1}$$

and the formula for the bending energy to

$$E^{\mathbf{bend}}_{[0,1]^2}(S) = \int_{[0,1]^2} \|\partial_1 \partial_1 S(\bar{u})\|^2 + 2\|\partial_1 \partial_2 S(\bar{u})\|^2 + \|\partial_2 \partial_2 S(\bar{u})\|^2 \, d\bar{u}. \tag{3.2}$$

This selection of the reference patch R allowed the precomputation of matrices $K^N_{[0,1]^2}$ for arbitrary valences N. One problem with this particular choice of the parametrization are entries of $K^N_{[0,1]^2}$ that are not well defined for $N \neq 4$. A few entries of the matrix $K_{[0,1]^2}$, which are computed as infinite sums, diverge.

It is known that $\int_S \kappa_1^2 + \kappa_2^2 \, d\omega_S < \infty$ for subdivision surfaces, even if S includes extraordinary points [RS01]. This means one can find suitable parametrizations, for which the integrals 3.1 and 3.2 won't diverge and hence the entries of K^N are well-defined. The parametrization of the subdivision patches over their characteristic maps [Rei95] is a prime candidate.

In this chapter we will introduce the characteristic map $\mathfrak{Ch}$ and parametrize the extraordinary patches with help of $\mathfrak{Ch}$. We compute the first and second partial derivatives of the Catmull-Clark subdivision patch S with regards to this parametrization. Using a scaling relation similar to the one employed in [HKD93] we show that the involved stiffness integrals converge. Finally formulas for the numerical evaluation of the stiffness matrices are given.

3.1 Using the Characteristic Map for Parametrization

The *characteristic map* $\mathfrak{Ch} : \mathbb{R}^2 \to \mathbb{R}^2$ of a subdivision surface near an extraordinary vertex is defined as

$$\mathfrak{Ch}(u_1, u_2) := \begin{bmatrix} \phi_2(u_1, u_2) \\ \phi_3(u_1, u_2) \end{bmatrix}. \tag{3.3}$$

The ϕ_i are the eigen functions in equation 2.15, corresponding to the sub-dominant eigen value λ. It is known that $\mathfrak{Ch}$ is regular and injective [Rei98].

The Catmull-Clark subdivision surface is only C^1 at extraordinary vertices. In most cases the second derivatives of the eigen functions ϕ_i diverge near $\bar{u} = (0, 0)$. This is one reason why we want to exclude a neighborhood of $(0, 0)$ from the integration domain and define the new domain as $L := [0, 1]^2 \setminus [0, \frac{1}{2})^2 \subset \mathbb{R}^2$. Without significant overlap the integration domain is split into a series of shrinking L-regions

$$[0, 1]^2 = \bigcup_{i=0}^{\infty} 2^{-i} \cdot L. \tag{3.4}$$

Now we define the image of L under the characteristic map as $L^{\mathfrak{Ch}} := \mathfrak{Ch}(L)$. Clearly

$$\mathfrak{Ch}([0, 1]^2) = \bigcup_{i=0}^{\infty} \lambda^i \cdot L^{\mathfrak{Ch}}. \tag{3.5}$$

The partial derivatives of the eigen functions considered parametrized over the characteristic map have some interesting properties

$$\partial_k \phi_i^{\mathfrak{Ch}}(\lambda^m \bar{v}) = \partial_k(\phi_i \circ \mathfrak{Ch}^{-1})(\lambda^m \bar{v}) \quad = \quad (\lambda_i/\lambda)^m \cdot \partial_k(\phi_i \circ \mathfrak{Ch}^{-1})(\bar{v}) \tag{3.6}$$

$$\partial_k \partial_l \phi_i^{\mathfrak{Ch}}(\lambda^m \bar{v}) = \partial_k \partial_l(\phi_i \circ \mathfrak{Ch}^{-1})(\lambda^m \bar{v}) \quad = \quad (\lambda_i/\lambda^2)^m \cdot \partial_k \partial_l(\phi_i \circ \mathfrak{Ch}^{-1})(\bar{v}). \tag{3.7}$$

To compute the membrane stiffness matrix $K_{\mathfrak{Ch}}^{\mathbf{memb}}$ one has to solve integrals

$$K_{ij} = \int_{\mathfrak{Ch}([0,1]^2)} \partial_1 \phi_i^{\mathfrak{Ch}}(\bar{v}) \cdot \partial_1 \phi_j^{\mathfrak{Ch}}(\bar{v}) + \partial_2 \phi_i^{\mathfrak{Ch}}(\bar{v}) \cdot \partial_2 \phi_j^{\mathfrak{Ch}}(\bar{v})\, d\bar{v}. \tag{3.8}$$

The eigen function $\phi_1^{\mathfrak{Ch}}(\bar{v}) = \textit{const}$. Hence if $i = 1$ or $j = 1$ the integral is zero. Now let $i, j > 1$.

$$\begin{aligned}
\int_{\mathfrak{Ch}([0,1]^2)} \partial_k \phi_i^{\mathfrak{Ch}}(\bar{v}) \cdot \partial_l \phi_j^{\mathfrak{Ch}}(\bar{v})\, d\bar{v} &= \sum_{m=0}^{\infty} \int_{\lambda^m \cdot L^{\mathfrak{Ch}}} \partial_k \phi_i^{\mathfrak{Ch}}(\bar{v}) \cdot \partial_l \phi_j^{\mathfrak{Ch}}(\bar{v})\, d\bar{v} \\
&= \sum_{m=0}^{\infty} \int_{L^{\mathfrak{Ch}}} (\lambda_i/\lambda)^m \cdot \partial_k \phi_i^{\mathfrak{Ch}}(\bar{v}) \cdot (\lambda_j/\lambda)^m \cdot \partial_l \phi_j^{\mathfrak{Ch}}(\bar{v}) \cdot \lambda^m \lambda^m\, d\bar{v} \\
&= (1 - \lambda_i \lambda_j)^{-1} \int_{L^{\mathfrak{Ch}}} \partial_k \phi_i^{\mathfrak{Ch}}(\bar{v}) \cdot \partial_l \phi_j^{\mathfrak{Ch}}(\bar{v})\, d\bar{v}
\end{aligned}$$

With the same arguments one can show for $i, j > 3$

$$\int_{\mathfrak{Ch}([0,1]^2)} \partial_k \partial_l \phi_i^{\mathfrak{Ch}}(\bar{v}) \cdot \partial_m \partial_n \phi_j^{\mathfrak{Ch}}(\bar{v})\ d\bar{v} = \left(1 - \frac{\lambda_i \lambda_j}{\lambda^2}\right)^{-1} \int_{L^{\mathfrak{Ch}}} \partial_k \partial_l \phi_i^{\mathfrak{Ch}}(\bar{v}) \cdot \partial_m \partial_n \phi_j^{\mathfrak{Ch}}(\bar{v})\ d\bar{v}. \tag{3.9}$$

The functions $\phi_2^{\mathfrak{Ch}}(\bar{v})$ and $\phi_3^{\mathfrak{Ch}}(\bar{v})$ are linear. Hence their second partial derivatives vanish and the left integral is zero if $i \leq 3$ or $j \leq 3$. For some subdivision schemes $\lambda_2 = \lambda_3 = \lambda_4$ [Rei98]. We haven't analyzed, if our derivation could be adapted to these schemes.

3.1.1 Numerical Evaluation of the Membrane Energy Integrals

We want to integrate

$$\int_{L^{\mathfrak{Ch}}} D\big(\phi_i^{\mathfrak{Ch}}(\bar{v})\big) \cdot D\big(\phi_j^{\mathfrak{Ch}}(\bar{v})\big)^T\, d\bar{v} \tag{3.10}$$

with $\phi_i^{\mathfrak{Ch}}(\bar{v}) := \phi_i \circ \mathfrak{Ch}^{-1}(\bar{v})$, $\phi_i^{\mathfrak{Ch}} : L^{\mathfrak{Ch}} \to \mathbb{R}$. It is not trivial to compute the inverse of the characteristic map $\mathfrak{Ch}^{-1}$, hence we want to avoid doing it. Now $D(\phi_i^{\mathfrak{Ch}})(\bar{v}) = D(\phi_i \circ \mathfrak{Ch}^{-1})(\bar{v}) = D(\phi_i)(\bar{v}) \cdot D(\mathfrak{Ch}^{-1})(\bar{v})$ and hence

$$\int_{L^{\mathfrak{Ch}}} D\big(\phi_i\big)\big(\mathfrak{Ch}^{-1}(\bar{v})\big) \cdot D(\mathfrak{Ch}^{-1})(\bar{v}) \cdot \Big(D\big(\phi_j\big)\big(\mathfrak{Ch}^{-1}(\bar{v})\big) \cdot D(\mathfrak{Ch}^{-1})(\bar{v})\Big)^T\, d\bar{v}. \tag{3.11}$$

Substituting $\bar{v} = \mathfrak{Ch}(\bar{u})$ we get

$$\int_L D\big(\phi_i\big)\Big(\mathfrak{Ch}^{-1}\big(\mathfrak{Ch}(\bar{u})\big)\Big) \cdot D(\mathfrak{Ch}^{-1})\big(\mathfrak{Ch}(\bar{u})\big) \cdot \bigg(D\big(\phi_j\big)\Big(\mathfrak{Ch}^{-1}\big(\mathfrak{Ch}(\bar{u})\big)\Big) \cdot D(\mathfrak{Ch}^{-1})\big(\mathfrak{Ch}(\bar{u})\big)\bigg)^T \cdot |J_{\mathfrak{Ch}}(\bar{u})|\, d\bar{u}$$

Using $D(\mathfrak{Ch}^{-1})\big(\mathfrak{Ch}(\bar{u})\big) = \big(D(\mathfrak{Ch})(\bar{u})\big)^{-1}$ we arrive at a formula that does not involve $\mathfrak{Ch}^{-1}$

$$\int_L D(\phi_i)(\bar{u}) \cdot \big(D(\mathfrak{Ch})(\bar{u})\big)^{-1} \cdot \Big(D(\phi_j)(\bar{u}) \cdot \big(D(\mathfrak{Ch})(\bar{u})\big)^{-1}\Big)^T \cdot |\mathbf{det}(D(\mathfrak{Ch})(\bar{u}))|\, d\bar{u}. \tag{3.12}$$

3.1.2 Numerical Evaluation of the Bending Energy Integrals

This section is a little technical. The change of integration variables from $L^{\mathfrak{Ch}}$ to L is computed. We use Einstein notation in most expressions.

To keep the size of the formulas small, we introduce the following conventions. The variables $\bar{u} = (u_1, u_2) \in L$ and $\bar{v} = (v_1, v_2) \in L^{\mathfrak{Ch}}$. The characteristic map and its inverse are abbreviated as $\mathfrak{v}(\bar{u}) = \mathfrak{Ch}(\bar{u}) : L \to L^{\mathfrak{Ch}}$ and $u(\bar{v}) = \mathfrak{Ch}^{-1}(\bar{v}) : L^{\mathfrak{Ch}} \to L$. Indices are used equivalently as sub- and superscripts. The colon is used to denote differentiation e.g. $\partial_k \partial_l f_j^i(\bar{u}) = f^i_{j,kl}(\bar{u})$.

From the last section we know that $Du = (D\mathfrak{v})^{-1}$. This is equivalent to $u^k_{,i} \cdot \mathfrak{v}^j_{,k} = \delta^j_i$ where δ is the Kronecker symbol. Equivalently

$$\begin{bmatrix} u^1_{,1} & u^2_{,1} \\ u^1_{,2} & u^2_{,2} \end{bmatrix} = \frac{1}{\mathfrak{v}^1_{,1} \cdot \mathfrak{v}^2_{,2} - \mathfrak{v}^2_{,1} \cdot \mathfrak{v}^1_{,2}} \begin{bmatrix} \mathfrak{v}^2_{,2} & -\mathfrak{v}^2_{,1} \\ -\mathfrak{v}^1_{,2} & \mathfrak{v}^1_{,1} \end{bmatrix} \tag{3.13}$$

allows us to express first partial derivatives of u in terms of first derivatives of $\mathfrak{v}$.

Now we compute the second partial derivatives $u^t_{,im}$ in terms of $\mathfrak{v}$.

$$\begin{aligned} \delta^j_i &= u^k_{,i} \cdot \mathfrak{v}^j_{,k} \\ 0 &= (u^k_{,i} \cdot \mathfrak{v}^j_{,k})_{,m} \\ 0 &= u^k_{,im} \cdot \mathfrak{v}^j_{,k} + u^k_{,i} \cdot \mathfrak{v}^j_{,kq} \cdot u^q_{,m} \end{aligned}$$

The last line is multiplied with $u^t_{,j}$ and the first product reordered

$$\begin{aligned} 0 &= u^t_{,j} \cdot \mathfrak{v}^j_{,k} \cdot u^k_{,im} + u^t_{,j} \cdot u^k_{,i} \cdot \mathfrak{v}^j_{,kq} \cdot u^q_{,m} \\ 0 &= \delta^t_k \cdot u^k_{,im} + u^t_{,j} \cdot u^k_{,i} \cdot \mathfrak{v}^j_{,kq} \cdot u^q_{,m} \\ 0 &= u^t_{,im} + u^t_{,j} \cdot u^k_{,i} \cdot \mathfrak{v}^j_{,kq} \cdot u^q_{,m} \end{aligned}$$

After appropriate renaming of the symbols we arrive at

$$u^t_{,im} = -\mathfrak{v}^c_{,ab} \cdot u^a_{,i} \cdot u^b_{,m} \cdot u^t_{,c}. \tag{3.14}$$

The right hand side involves only $\mathfrak{v}^c_{,ab}$ and first partial derivatives of u. We already know how to handle the latter ones with equation 3.13.

The problems that need to be solved to compute the stiffness matrix are integrals of the form

$$\int_{L^{\mathfrak{Ch}}} \phi^{\mathfrak{Ch}}_{k,ij}(\bar{v}) \cdot \phi^{\mathfrak{Ch}}_{l,ij}(\bar{v})\ d\bar{v}. \tag{3.15}$$

Because of the growth of the terms involved we restrict our attention to the transformations necessary for computing

$$\begin{aligned} \int_{L^{\mathfrak{Ch}}} \phi^{\mathfrak{Ch}}_{,ij}(\bar{v})\ d\bar{v} &= \int_{L^{\mathfrak{v}}} (\phi \circ \mathfrak{v})_{,ij}(\bar{v})\ d\bar{v} \\ &= \int_{L^{\mathfrak{v}}} (\phi_{,k}(u(\bar{v})) \cdot u^k_{,j}(\bar{v}))_{,i}\ d\bar{v} \\ &= \int_{L^{\mathfrak{v}}} \phi_{,ks}(u(\bar{v})) \cdot u^s_{,i}(\bar{v}) \cdot u^k_{,j}(\bar{v}) + \phi_{,k}(u(\bar{v})) \cdot u^k_{,ji}(\bar{v})\ d\bar{v}. \end{aligned}$$

After substituting $\bar{v} = \mathfrak{v}(\bar{u})$ we get

$$\int_{L} \Big(\phi_{,ks}(\bar{u}) \cdot u^{s}_{,i}\big(\mathfrak{v}(\bar{u})\big) \cdot u^{k}_{,j}\big(\mathfrak{v}(\bar{u})\big) + \phi_{,k}(\bar{u}) \cdot u^{k}_{,ji}\big(\mathfrak{v}(\bar{u})\big)\Big) \cdot |J_{\mathfrak{v}}(\bar{u})| \, d\bar{u}. \tag{3.16}$$

The final formula, which depends only on derivatives of ϕ and $\mathfrak{v}$ but not its inverses, is obtained by replacing $u^{s}_{,i}$ and $u^{k}_{,ji}$ with help of equations 3.13 and 3.14.

All functions involved in this final formula are eigen functions of the Catmull-Clark subdivision patch parametrized over L. The eigen functions are C^2 on an open neighborhood including L. Because of the regularity of $\mathfrak{Ch}$ [Rei95] the determinant of $\mathfrak{Ch}$ does not vanish. As a consequence the integrand of 3.16 (and also the integrand of 3.15) is smooth and bounded on L. Hence it is best computed using high order quadrature rules.

Chapter 4

First Order Data Dependent Energies

The bending energy of a sphere $\mathbf{Sp}(r)$ with radius r is $\int_S \kappa_1^2 + \kappa_2^2 d\omega_S = \int_S \frac{1}{r^2} + \frac{1}{r^2} d\omega_S = \frac{2}{r^2} \cdot 4\pi r^2 = 8\pi$. The energy $E^{\mathbf{bend}}_{\mathfrak{Ch}}(\mathbf{Sp}(r))$ computed with the data independent characteristic map parametrization of the previous chapter is proportional to r^2. This behavior is clearly wrong. It might not matter in some applications, especially if the surface is scaled globally and uniformly. But if the scaling is non-uniform in different coordinate directions or patches vary significantly in size, then the estimate $E^{\mathbf{bend}}_{\mathfrak{Ch}}(S)$ can be made locally arbitrarily bad compared to $E^{\mathbf{bend}}_{S}(S)$.

This observation motivates the scaling of the reference surface patch according to the dimensions of the original surface patch. This makes the energy $E_R(S)$ less dependent on the parametrization of S [GLW96].

4.1 Linear Transformation of the Characteristic Map

Let $\mathfrak{S}\mathfrak{Ch}(\bar{u})$ be a linearly transformed version of the characteristic map $\mathfrak{Ch}(\bar{u})$. The dependent energy functional is invariant to translation. This allows moving $\mathfrak{S}\mathfrak{Ch}(\bar{u})$ to the origin of the coordinate system and writing the translated version of $\mathfrak{S}\mathfrak{Ch}(\bar{u})$ as

$$\mathfrak{S}\mathfrak{Ch}(\bar{u}) = W \cdot \mathfrak{Ch}(\bar{u}) = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix} \cdot \begin{bmatrix} \phi_2(\bar{u}) \\ \phi_3(\bar{u}) \end{bmatrix}, \tag{4.1}$$

The characteristic map $\mathfrak{Ch}(\bar{u})$ is also planar. The rotation invariance of the data dependent energy functionals allows rotating it into the x-y plane, without changing the energies. Ignoring the $z-$component of the translated and rotated version of $\mathfrak{S}\mathfrak{Ch}(\bar{u})$ the number of free parameters of the reference surface is reduced to

$$\mathfrak{S}\tilde{\mathfrak{C}}\mathfrak{h}(\bar{u}) = \tilde{W} \cdot \mathfrak{Ch}(\bar{u}) = \begin{bmatrix} s_x & s_{xy} \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} \phi_2(\bar{u}) \\ \phi_3(\bar{u}) \end{bmatrix}, \tag{4.2}$$

with $s_x, s_{xy}, s_y \in \mathbb{R}$. It is easily understood, that s_x, s_y are *scaling factors* of the characteristic map and s_{xy} is a *shearing term.*

4.1.1 Influence of the Map W

It is simple to verify, that $\partial_i \mathfrak{SCh}(\bar{u}) = W \cdot \partial_i \mathfrak{Ch}(\bar{u})$. For reasons that will be obvious later, we set $V := W^T W =: \left[\begin{smallmatrix} E & F \\ F & G \end{smallmatrix}\right]$. Clearly $V^{-1} = \frac{1}{EG - F^2} \cdot \left[\begin{smallmatrix} G & -F \\ -F & E \end{smallmatrix}\right]$. Now it follows, that

$$\begin{aligned} J_{\mathfrak{SCh}} &= |\mathbf{det}(\tilde{W})| \cdot J_{\mathfrak{Ch}} = |\mathbf{det}(V)|^{\frac{1}{2}} \cdot J_{\mathfrak{Ch}} \quad \text{and} && (4.3) \\ \mathbf{I}_{\mathfrak{SCh}} &= D\mathfrak{Ch}(\bar{u})^T \cdot W^T \cdot W \cdot D\mathfrak{Ch}(\bar{u}). && (4.4) \end{aligned}$$

From the latter equation we see, that the inverse of $\mathbf{I}_{\mathfrak{SCh}}$ is given by

$$\mathbf{I}_{\mathfrak{SCh}}^{-1} = \big(D\mathfrak{Ch}(\bar{u})\big)^{-1} \cdot (W^T \cdot W)^{-1} \cdot \big(D\mathfrak{Ch}(\bar{u})\big)^{-T}. \tag{4.5}$$

It is somewhat tedious, but straight forward, to check that $\Gamma^i_{\mathfrak{SCh}} = \Gamma^i_{\mathfrak{Ch}}$. Hence we have

$$\begin{aligned} \mathbf{Hess}_{\mathfrak{SCh}}(h) &= I_{\mathfrak{SCh}}^{-1} \cdot \big(\mathbf{Hess}(h) - (\partial_1 h \Gamma^1 + \partial_2 h \Gamma^2)\big) && (4.6) \\ &= I_{\mathfrak{SCh}}^{-1} \cdot Q(h, \mathfrak{Ch}), && (4.7) \end{aligned}$$

where $Q(h, \mathfrak{Ch})$ is some symmetric matrix depending on $h(\bar{u})$ and $\mathfrak{Ch}(\bar{u})$, but not on W.

4.2 Precomputing the Membrane Energy Integrals

Let $\bar{u} \in [0,1]^2$ and $\bar{v} \in \mathfrak{SCh}([0,1]^2)$. The first order data dependent membrane energy stiffness matrix integral

$$K_{ij}^{\mathbf{memb}} = \int_{\mathfrak{SCh}([0,1]^2)} D\big(\phi_i^{\mathfrak{SCh}}(\bar{v})\big) \cdot D\big(\phi_j^{\mathfrak{SCh}}(\bar{v})\big)^T \, d\bar{v} \tag{4.8}$$

can be transformed to

$$\int_{[0,1]^2} D(\phi_i)(\bar{u}) \cdot \big(D\mathfrak{Ch}(\bar{u})\big)^{-1} \cdot \tilde{W}^{-1} \cdot \Big(D(\phi_j)(\bar{u}) \cdot \big(D\mathfrak{Ch}(\bar{u})\big)^{-1} \cdot \tilde{W}^{-1}\Big)^T \cdot |\mathbf{det}(\tilde{W})| \cdot |J_{\mathfrak{Ch}}(\bar{u})| \, d\bar{u}$$

$$\begin{aligned}
&= |\mathbf{det}(\tilde{W})| \cdot \int_{[0,1]^2} \mathrm{D}(\phi_i)(\bar{u}) \cdot \big(\mathrm{D}\mathfrak{Ch}(\bar{u})\big)^{-1} \cdot \tilde{W}^{-1} \cdot \tilde{W}^{-\mathsf{T}} \cdot \big(\mathrm{D}\mathfrak{Ch}(\bar{u})\big)^{-\mathsf{T}} \cdot \mathrm{D}(\phi_j)(\bar{u})^{\mathsf{T}} \cdot |\mathrm{J}_{\mathfrak{Ch}}(\bar{u})| \; d\bar{u} \\
&= |\mathbf{det}(V)|^{\frac{1}{2}} \cdot \int_{[0,1]^2} \mathrm{D}(\phi_i)(\bar{u}) \cdot \big(\mathrm{D}\mathfrak{Ch}(\bar{u})\big)^{-1} \cdot V^{-1} \cdot \big(\mathrm{D}\mathfrak{Ch}(\bar{u})\big)^{-\mathsf{T}} \cdot \mathrm{D}(\phi_j)(\bar{u})^{\mathsf{T}} \cdot |\mathrm{J}_{\mathfrak{Ch}}(\bar{u})| \; d\bar{u} \\
&= \frac{|\mathbf{det}(V)|^{\frac{1}{2}}}{\mathbf{det}(V)} \cdot \int_{[0,1]^2} \mathrm{D}(\phi_i)(\bar{u}) \cdot \big(\mathrm{D}\mathfrak{Ch}(\bar{u})\big)^{-1} \cdot \left[\begin{smallmatrix} G & -F \\ -F & E \end{smallmatrix}\right] \cdot \big(\mathrm{D}\mathfrak{Ch}(\bar{u})\big)^{-\mathsf{T}} \cdot \mathrm{D}(\phi_j)(\bar{u})^{\mathsf{T}} \cdot |\mathrm{J}_{\mathfrak{Ch}}(\bar{u})| \; d\bar{u} \\
&= |(EG - F^2)|^{-\frac{1}{2}} \cdot \Big(E \int_{[0,1]^2} f^{E}_{ij}(\bar{u}) \; d\bar{u} + F \int_{[0,1]^2} f^{F}_{ij}(\bar{u}) \; d\bar{u} + G \int_{[0,1]^2} f^{G}_{ij}(\bar{u}) \; d\bar{u} \Big) \\
&= (EG - F^2)^{-0.5} \cdot \big(E \cdot K^{E} + F \cdot K^{F} + G \cdot K^{G} \big)_{ij}
\end{aligned}$$

where the functions $f^{E}_{ij}(\bar{u})$, $f^{F}_{ij}(\bar{u})$ and $f^{G}_{ij}(\bar{u})$ do not depend on the choice of the parameters E, F, G (or equivalently on the choice of s_x, s_{xy} and s_y). This means $K^{\mathbf{memb}}_{\mathfrak{S}\mathfrak{Ch}}$ can be evaluated by precomputing three matrices K^{E}, K^{F} and K^{G} and scaling them later as needed.

Comparing the first of the equations with equation 3.11 it is not hard to see, that the integrals above can be computed exactly the same way as in the non data dependent case as sums of integrals over $L-$regions.

4.3 Evaluating the Bending Energy Integrals

As in the unscaled case, we try to avoid numerical integration close to the origin, where the integrand has poles. By definition of $\mathfrak{S}\mathfrak{Ch}$ we have $\mathfrak{S}\mathfrak{Ch}(2^{-n}\bar{u}) = W \cdot \mathfrak{Ch}(2^{-n}\bar{u}) = \big(W \cdot \left[\begin{smallmatrix} \lambda & 0 \\ 0 & \lambda \end{smallmatrix}\right]^{n}\big) \cdot \mathfrak{Ch}(\bar{u})$. This leads to the idea that the necessary change of the integration domain from $2^{-n}L$ to L is governed by the rules derived for the change of $\mathbf{Hess}_{\mathfrak{S}\mathfrak{Ch}}$ under application of a linear map and the scaling relations of the eigen functions. Recall, that

$$\begin{aligned}
K_{ij} = & \quad \int_{[0,1]^2} \mathbf{trace}\Big(\mathbf{Hess}_{\mathfrak{S}\mathfrak{Ch}}\big(\phi_i(\bar{v})\big) \cdot \mathbf{Hess}_{\mathfrak{S}\mathfrak{Ch}}\big(\phi_j(\bar{v})\big)^{\mathsf{T}} \Big) \cdot |\mathrm{J}_{\mathfrak{S}\mathfrak{Ch}}(\bar{v})| \; d\bar{v} \\
= \sum_{n=0}^{\infty} & \quad \int_{2^{-n} \cdot L} \mathbf{trace}\Big(\mathbf{Hess}_{\mathfrak{S}\mathfrak{Ch}}\big(\phi_i(\bar{v})\big) \cdot \mathbf{Hess}_{\mathfrak{S}\mathfrak{Ch}}\big(\phi_j(\bar{v})\big)^{\mathsf{T}} \Big) \cdot |\mathrm{J}_{\mathfrak{S}\mathfrak{Ch}}(\bar{v})| \; d\bar{v}
\end{aligned}$$

We pick one of the integrals in the sum and fix n. Now let us examine the effects of the substitution $\bar{v} = 2^{-n} \cdot \bar{u}$ on the terms in $\mathbf{Hess}_{\mathfrak{S}\mathfrak{Ch}}(\phi_j(\bar{v}))$ in detail.

$$\mathbf{Hess}_{\mathfrak{S}\mathfrak{Ch}}(\phi_j(\bar{v})) = \mathrm{I}^{-1}_{\mathfrak{S}\mathfrak{Ch}}(\bar{v}) \cdot \Big(\mathbf{Hess}(\phi_j(\bar{v})) - (\partial_1 \phi_j(\bar{v}) \Gamma^1(\bar{v}) + \partial_2 \phi_j(\bar{v}) \Gamma^2(\bar{v})) \Big)$$

The derivatives of the eigen functions scale as

$$\begin{aligned} \partial_k \phi_i(2^{-n}\bar{u}) &= 2^n \lambda_i^n \cdot \partial_k \phi_i(\bar{u}) \quad \text{and} && (4.9) \\ \partial_k \partial_l \phi_i(2^{-n}\bar{u}) &= 2^{2n} \lambda_i^n \cdot \partial_k \partial_l \phi_i(\bar{u}). && (4.10) \end{aligned}$$

We can immediately write $\mathbf{Hess}(\phi_j)(2^{-n}\bar{u}) = 2^{2n}\lambda_j^n \cdot \mathbf{Hess}(\phi_j)(\bar{u})$. Application of the first of these equations to the definition of the first fundamental form yields $\mathbf{I}^{-1}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(2^{-n}\bar{u}) = (2\lambda)^{-2n} \cdot \mathbf{I}^{-1}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{u})$. From this we reason $\Gamma^k_{ij}(2^{-n}\bar{u}) = g^{kl} < \partial_i\partial_j\mathfrak{C}\mathfrak{h}, \partial_l\mathfrak{C}\mathfrak{h} >= 2^n \cdot \Gamma^k_{ij}(\bar{u})$. Combining these three results we get $\mathbf{Hess}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\phi_j)(2^{-n}\bar{u}) = \left(\frac{\lambda_j}{\lambda^2}\right)^n \cdot \mathbf{Hess}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\phi_j)(\bar{u})$. The Jacobian scales as before $J_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(2^{-n}\bar{u}) = \lambda^{2n} \cdot J_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{u})$.

Now we have everything together to finalize the change of integration variables

$$\begin{aligned} &\int_{2^{-n}\cdot L} \mathbf{trace}\Big(\mathbf{Hess}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}\big(\phi_i(\bar{v})\big) \cdot \mathbf{Hess}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}\big(\phi_j(\bar{v})\big)^T\Big) \cdot |J_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{v})| \, d\bar{v} = \\ &\int_L \mathbf{trace}\Big(\mathbf{Hess}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\phi_i(\bar{u})) \cdot \Big(\frac{\lambda_i}{\lambda^2}\Big)^n \cdot \Big(\frac{\lambda_j}{\lambda^2}\Big)^n \cdot \mathbf{Hess}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\phi_j(\bar{u}))^T\Big) \cdot \lambda^{2n} \cdot |J_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{u})| \, d\bar{u} = \\ &\Big(\frac{\lambda_i\lambda_j}{\lambda^2}\Big)^n \cdot \int_L \mathbf{trace}\Big(\mathbf{Hess}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\phi_i(\bar{u})) \cdot \mathbf{Hess}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\phi_j(\bar{u}))^T\Big) \cdot |J_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{u})| \, d\bar{u}. \qquad (4.11) \end{aligned}$$

The last formula shows the same scaling relation as the simple bending integral 3.9.

4.3.1 Precomputing the Bending Energy Integrals

Now we want to find an efficient way to assemble the matrix $K_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}$. We have

$$\begin{aligned} K^{\mathfrak{S}\mathfrak{C}\mathfrak{h}}_{ij} &= \Big(1 - \frac{\lambda_i\lambda_j}{\lambda^2}\Big)^{-1} \int_L \mathbf{trace}\Big(\mathbf{Hess}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\phi_i) \cdot \mathbf{Hess}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\phi_j)^T\Big)^T d\omega_{\mathfrak{S}\mathfrak{C}\mathfrak{h}} \\ &= \Big(1 - \frac{\lambda_i\lambda_j}{\lambda^2}\Big)^{-1} \int_L \mathbf{trace}\Big(Q(\phi_i, \mathfrak{C}\mathfrak{h}) I^{-1}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}} \cdot I^{-1}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}} \cdot Q(\phi_j, \mathfrak{C}\mathfrak{h})\Big) \cdot |\mathbf{det}(V)|^{\frac{1}{2}} \, d\omega_{\mathfrak{C}\mathfrak{h}} \\ &= \Big(1 - \frac{\lambda_i\lambda_j}{\lambda^2}\Big)^{-1} \cdot |\mathbf{det}(V)|^{\frac{1}{2}} \cdot \int_L \mathbf{trace}\Big(Q(\phi_i, \mathfrak{C}\mathfrak{h}) D\mathfrak{C}\mathfrak{h}^{-1} V^{-1} D\mathfrak{C}\mathfrak{h}^{-1} \cdot \\ &\qquad\qquad D\mathfrak{C}\mathfrak{h}^{-1} V^{-1} D\mathfrak{C}\mathfrak{h}^{-1} Q(\phi_j, \mathfrak{C}\mathfrak{h})\Big) \, d\omega_{\mathfrak{C}\mathfrak{h}} \end{aligned}$$

Because $V^{-1} = \frac{1}{EG - F^2} \cdot \left[\begin{smallmatrix} G & -F \\ -F & E \end{smallmatrix}\right]$ the last integral can be rewritten as

$$\begin{aligned} K^{\mathfrak{S}\mathfrak{C}\mathfrak{h}}_{ij} &= \frac{(EG - F^2)^{\frac{1}{2}}}{(EG - F^2)^2} \cdot \Big(1 - \frac{\lambda_i\lambda_j}{\lambda^2}\Big)^{-1} \int_L E^2 \cdot f^{EE}_{ij}(\bar{u}) + EF \cdot f^{EF}_{ij}(\bar{u}) + EG \cdot f^{EG}_{ij}(\bar{u}) + \\ &\qquad F^2 \cdot f^{FF}_{ij}(\bar{u}) + FG \cdot f^{FG}_{ij}(\bar{u}) + G^2 \cdot f^{GG}_{ij}(\bar{u}) \, d\bar{u}, \end{aligned}$$

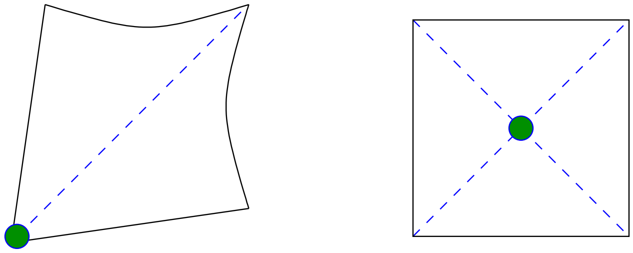


Figure 4.1: Selecting sampling locations for E, F and G with respect to axial symmetries of the extraordinary (left) and the regular patch (right).

where each of the functions f^{xx} is independent of the parameters E, F and G. (This transformation is best done using an computer algebra system.) With $xx \in \{EE, EF, EG, FF, FG, GG\}$ this reads as

$$K_{ij}^{\mathfrak{S}\mathfrak{Ch}} = \sum_{xx} xx \cdot (EG - F^2)^{-1.5} \cdot \Big(\big(1 - \frac{\lambda_i \lambda_j}{\lambda^2}\big)^{-1} \int_{L^u} f_{ij}^{xx}(\bar{u})\, d\bar{u} \Big). \tag{4.12}$$

Finally this formula is used to define 6 auxiliary matrices

$$K_{ij}^{xx} = \Big(1 - \frac{\lambda_i \lambda_j}{\lambda^2}\Big)^{-1} \int_{L^u} f_{ij}^{xx}(\bar{u})\, d\bar{u} \tag{4.13}$$

that are independent of W and are computed numerically with high precision in an offline process. The 6 corresponding scaling factors are defined as $c_{xx} = xx \cdot (EG - F^2)^{-1.5}$.

4.4 Choosing the Parameters E, F and G

We have already found an interpretation for s_x, s_{xy} and s_y as scaling and shearing terms of the characteristic map. To compute E, F and G one could estimate these parameters from the subdivision patch S using geometric properties.

There is another approach to obtain - hopefully good - values for E, F and G. It is based on the observation, that one can fix a parameter value $\bar{u}_s \in [0,1]^2$ and use the map W to translate and rotate the reference patch $\mathfrak{S}\mathfrak{Ch}(\bar{u}_s)$ into the tangential plane of the subdivision patch $S(\bar{u}_s)$. The map W can furthermore be used to match the first partial derivatives $\partial_i S(\bar{u}_s) = \partial_i \mathfrak{S}\mathfrak{Ch}(\bar{u}_s)$, and hence the first fundamental forms $\mathbf{I}_S(\bar{u}_s) = \mathbf{I}_{\mathfrak{S}\mathfrak{Ch}}(\bar{u}_s)$ of both patches at this particular point.

Now how should one chose $\bar{u}_s$? The partial derivatives $\partial_i S$ and the first fundamental form $\mathbf{I}_S$ are continuous on $[0,1]^2$. Hence any point $\bar{u}_s \in [0,1]^2$ is a candidate for estimating E, F and G.

The image of the characteristic map $\mathfrak{Ch}(\bar{u})$ has one axis of symmetry (compare with figure 4.1). It seems to make sense to restrict the choice of $\bar{u}_s$ such that its image is on this axis. Hence $\bar{u}_s = (t,t)$ with $t \in [0,1]$. Now in the regular case the choice of $\bar{u}_s$ should be symmetrical with respect to all four corner vertices. The center $\bar{u}_s = (0.5, 0.5)$ of the regular patch is the only location to fulfill this requirement. Unfortunately irregular patches don't have such a unique point and it is not clear which additional requirement would lead to an optimal choice of t.

Now recall equation 2.15. For $\bar{u} = (0,0)$ the partial derivatives of S are computed as

$$\partial_i S(0,0) = \partial_i \sum_{i=1}^{M} C_i \cdot \phi_i(0,0) = C_2 \cdot \partial_i \phi_2(0,0) + C_3 \cdot \partial_i \phi_3(0,0). \tag{4.14}$$

This is a very convenient and easily evaluated formula. The linear dependence of C_2 and C_3 on the control points P_i of the extraordinary patch also simplifies the computation of the derivatives of the energy functional $E_{\mathfrak{SCh}}(\bar{X})$ discussed in chapter 5.

4.5 Final Formulas for the Energies

The evaluation of the first order data dependent membrane energy takes the form

$$E^{\mathbf{memb}}_{\mathfrak{SCh}}(P) = P^T \cdot \Big(c_E(P)K^E + c_F(P)K^F + c_G(P)K^G\Big) \cdot P, \tag{4.15}$$

where $c_E = E \cdot (EG - F^2)^{-0.5}$, $c_F = F \cdot (EG - F^2)^{-0.5}$ and $c_G = G \cdot (EG - F^2)^{-0.5}$.

The first order data dependent bending energy is computed as

$$E^{\mathbf{bend}}_{\mathfrak{SCh}}(P) = P^T \cdot \Big(c_{EE}(P)K^{EE} + c_{EF}(P)K^{EF} + c_{EG}(P)K^{EG} + c_{FF}(P)K^{FF} + c_{FG}(P)K^{FG} + c_{GG}(P)K^{GG}\Big) \cdot P. \tag{4.16}$$

where $c_{EE} = EE \cdot (EG - F^2)^{-1.5}$, $c_{EF} = EF \cdot (EG - F^2)^{-1.5}$ etc.

All matrices are precomputed and the scaling factors are easily evaluated. Hence the evaluation of the energies is dominated by memory fetch operations accessing the matrices. The run-times for an efficient implementation will be at most 3× resp. 6× the time for evaluating the simple, not data dependent, energies.

Chapter 5

Derivatives of the Energy Functionals

We want to regard the control points P of the patch S as variables. For clarity we use the identifier $\bar{\mathsf{X}}$ for unconstrained points, while we reserve the variable P for a specific choice of the control points.

$$\mathsf{E}(\bar{\mathsf{X}}) = \bar{\mathsf{X}}^{\mathsf{T}} \cdot \bar{\mathsf{K}} \cdot \bar{\mathsf{X}} = \begin{bmatrix} \mathsf{P}_x & \mathsf{P}_y & \mathsf{P}_z \end{bmatrix} \cdot \begin{bmatrix} \mathsf{K} & 0 & 0 \\ 0 & \mathsf{K} & 0 \\ 0 & 0 & \mathsf{K} \end{bmatrix} \cdot \begin{bmatrix} \mathsf{P}_x \\ \mathsf{P}_y \\ \mathsf{P}_z \end{bmatrix} \tag{5.1}$$

The matrix $\bar{\mathsf{K}}$ has dimension $3\mathsf{M} \times 3\mathsf{M}$, where M is the number of basis functions needed to represent the surface.

5.1 Derivatives of $\mathsf{E}_{\mathfrak{Ch}}(\bar{\mathsf{X}})$

Because $\bar{\mathsf{K}}$ is independent of the point-variables $\bar{\mathsf{X}}$, equation 5.1 is a multivariate polynomial in $\bar{\mathsf{X}}_i$ of degree 2. Hence the gradient of $\mathsf{E}_{\mathfrak{Ch}}(\bar{\mathsf{X}})$ is

$$\mathbf{grad}(\mathsf{E}_{\mathfrak{Ch}}(\bar{\mathsf{X}})) = 2 \cdot \bar{\mathsf{X}}^{\mathsf{T}} \cdot \bar{\mathsf{K}} \tag{5.2}$$

and the Hessian

$$\mathbf{Hess}(\mathsf{E}_{\mathfrak{Ch}}(\bar{\mathsf{X}})) = 2 \cdot \bar{\mathsf{K}}. \tag{5.3}$$

The gradient as well as the Hessian are trivial to compute. The complexity is $\mathsf{O}(\mathsf{M}^2)$. This is about as expensive as a single function evaluation of $\mathsf{E}_{\mathfrak{Ch}}(\bar{\mathsf{X}})$.

5.2 Derivatives of $\mathsf{E}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{\mathsf{X}})$

The matrices $\bar{\mathsf{K}}(\bar{\mathsf{X}})$ depend on the data $\bar{\mathsf{X}}$. This makes **grad**$(\mathsf{E}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{\mathsf{X}}))$ nonlinear and there is no hope that **Hess**$(\mathsf{E}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{\mathsf{X}}))$ is constant.

A classical solution to our problem is the computation of numerical finite differences to approximate the partial derivatives of $\mathsf{E}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{\mathsf{X}})$. This method has two drawbacks. The first is speed: One has to evaluate $\mathsf{E}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}$ for computing the gradient $\mathsf{O}(\mathsf{M})$ and for the Hessian $\mathsf{O}(\mathsf{M}^2)$ times. Because each evaluation of $\mathsf{E}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}$ is done in $\mathsf{O}(\mathsf{M}^2)$ steps, the computation (especially of the Hessian) is very expensive. The second problem is accuracy: Even if $\mathsf{E}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}$ is computed as in our implementation with about $10...14$ valid decimal digits, the finite difference gradient approximation will have only about $6...7$ and the Hessian about $3...5$ valid leading digits. In addition to that good numerical differencing code is not trivial to write.

A modern approach for computing derivatives of a function is given by automatic differentiation techniques. The general idea is based on the observation, that any formula is computed using only a finite set of unary or binary functions. This allows the automatic analysis of the data flow and even the most complex formula is differentiated using the chain rule. While the author had none of these tools on hand, it is reported in [NW99] that two general methods are available: one method computes derivatives very fast, but uses potentially huge amounts of memory, while the other method conserves memory, but is relatively slow.

In the case of $\mathsf{E}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{\mathsf{X}})$ the non-linearity is quite simple to evaluate algebraically. It turns out that it is possible to differentiate $\mathsf{E}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{\mathsf{X}})$ manually and implement a direct computation of its derivatives with little required additional memory (only $\mathsf{O}(\mathsf{M})$) and no need of computing complex quantities multiple times.

For simplicity let us focus on $c_{\mathsf{EF}}(\bar{\mathsf{X}}) \cdot \bar{\mathsf{X}}^{\mathsf{T}} \cdot \bar{\mathsf{K}}_{\mathsf{EF}} \cdot \bar{\mathsf{X}}$. The gradient is computed as

$$\begin{aligned} \mathbf{grad}(\mathsf{E}^{\mathsf{EF}}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{\mathsf{X}}))_k &= \partial_k\Big(c_{\mathsf{EF}}(\bar{\mathsf{X}}) \sum_{m,n} \bar{\mathsf{K}}^{\mathsf{EF}}_{mn}\bar{\mathsf{X}}_m\bar{\mathsf{X}}_n\Big) && (5.4)\\ &= \partial_k c_{\mathsf{EF}}(\bar{\mathsf{X}}) \cdot \sum_{m,n} \bar{\mathsf{K}}^{\mathsf{EF}}_{mn}\bar{\mathsf{X}}_m\bar{\mathsf{X}}_n + c_{\mathsf{EF}}(\bar{\mathsf{X}}) \cdot 2 \cdot \sum_{m} \bar{\mathsf{K}}^{\mathsf{EF}}_{mk}\bar{\mathsf{X}}_m. && (5.5) \end{aligned}$$

For the Hessian we get

$$\begin{aligned}
\mathbf{Hess}(E^{EF}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{X}))_{lk} &= \partial_l \partial_k \Big(c_{EF}(\bar{X}) \sum_{m,n} \bar{K}^{EF}_{mn} \bar{X}_m \bar{X}_n \Big) && (5.6) \\
&= \partial_l \Big(\partial_k c_{EF}(\bar{X}) \cdot \sum_{m,n} \bar{K}^{EF}_{mn} \bar{X}_m \bar{X}_n + c_{EF}(\bar{X}) \cdot 2 \cdot \sum_{m} \bar{K}^{EF}_{mk} \bar{X}_m \Big) && (5.7) \\
&= \partial_l \partial_k c_{EF}(\bar{X}) \cdot \sum_{m,n} \bar{K}^{EF}_{mn} \bar{X}_m \bar{X}_n + \partial_l c_{EF}(\bar{X}) \cdot 2 \cdot \sum_{m} \bar{K}^{EF}_{mk} \bar{X}_m + && (5.8) \\
&\quad \partial_k c_{EF}(\bar{X}) \cdot 2 \cdot \sum_{m} \bar{K}^{EF}_{ml} \bar{X}_m + c_{EF}(\bar{X}) \cdot 2 \cdot \bar{K}^{EF}_{lk}. && (5.9)
\end{aligned}$$

Evaluating the derivatives of $c_{EF}(\bar{X}) = E(\bar{X})F(\bar{X}) \cdot \big(E(\bar{X})G(\bar{X}) - F(\bar{X})^2\big)^{-1.5}$ is the trickiest part. This is done by recursive application of the chain- and product rules down to the level of $\partial_k E(\bar{X})$, $\partial_k F(\bar{X})$ and $\partial_k G(\bar{X})$. All intermediate derivatives ∂_k are stored in arrays of size $3M$. No intermediate second derivative $\partial_l \partial_k$ needs to be stored or computed twice.

Experiments show, that on average the evaluation of the gradient $\mathbf{grad}(E_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{X}))$ with its $3M$ entries is only 1.5 times as expensive as a single function evaluation $E_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{X})$. The evaluation of the complete Hessian with $3M \times 3M$ entries is only $10...15$ times as expensive as a single function evaluation. (With finite differences and valence 4 the Hessian would have been about 2500 times as expensive as a single function evaluation achieving much less accuracy!)

The Hessian of the data dependent energy functional is not block-diagonal as the Hessian of the simple energies in equation 5.3. The elements off-diagonal are mostly nonzero. This is an indicator, that p_x, p_y, p_z interact with help of $c_{xx}(\bar{X})$. This again has consequences for the separability of equation systems based on the Hessian of $E_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(\bar{X})$.

Chapter 6

Experiments

For the simple operators of chapter 3 we can assume $\mathsf{E} = 1$, $\mathsf{F} = 0$ and $\mathsf{G} = 1$. As a first numerical test of our implementation we have verified, that

$$\begin{aligned} \mathsf{K}^{\mathbf{memb}}_{\mathfrak{Ch}} &= \mathsf{K}^{\mathsf{E}} + \mathsf{K}^{\mathsf{G}} \quad \text{and} \\ \mathsf{K}^{\mathbf{bend}}_{\mathfrak{Ch}} &= \mathsf{K}^{\mathsf{EE}} + \mathsf{K}^{\mathsf{EG}} + \mathsf{K}^{\mathsf{GG}}. \end{aligned}$$

In the rest of this chapter we will examine and compare properties of the simple and first order data dependent energy operators of chapters 3 and 4. We selected two primary models, a torus, which consists only of regular patches, and a subdivided cone, having irregular patches of valence 3 and 7.

The question that seems to be of most interest, is how robust the operators are with respect to parametrization changes. This is examined using global linear transformation and repeated subdivision. At the end of this chapter the gradients and Hessians of the bending energy operators are compared.

6.1 Linear Transformation of Meshes

In this section we want to analyze the behavior of the energy operators, if each of the control points of a mesh is transformed linearly with the same map. The transformations considered are the identity map I, a uniform scale $\mathsf{U} = \mathbf{diag}(10, 10, 10)$ in each coordinate direction and finally the more complex transformation

$$\mathsf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 1 \\ 0 & 0 & 0.5 \end{bmatrix}, \tag{6.1}$$

which involves different scales for each of the coordinate directions plus a shearing term.

Figures 6.1 and 6.2 display the relative errors of the simple and data dependent functionals, if compared with their corresponding exact functionals. We make the following observations: The data dependent membrane energy functional $\mathsf{E}^{\mathbf{memb}}_{\mathfrak{SCh}}(\mathsf{S})$ models the exact functional $\mathsf{E}^{\mathbf{memb}}_{\mathsf{S}}(\mathsf{S})$ very accurately for all three transformations. The simple bending energy $\mathsf{E}^{\mathbf{bend}}_{\mathfrak{Ch}}(\mathsf{S})$ is not a good model in any of the three cases. The simple membrane energy $\mathsf{E}^{\mathbf{memb}}_{\mathfrak{Ch}}(\mathsf{S})$ and the data dependent energy $\mathsf{E}^{\mathbf{bend}}_{\mathfrak{SCh}}(\mathsf{S})$ seem to model the energies of the original and uniformly scaled meshes well, while doing a relatively poor job on the meshes transformed by the map M.

The reason for the relatively good behavior of $\mathsf{E}^{\mathbf{memb}}_{\mathfrak{Ch}}(\mathsf{S})$ with respect to the uniform scaling U is rooted in the inherent scaling relations of both the quadratic form $\mathsf{E}^{\mathbf{memb}}_{\mathfrak{Ch}}(\mathsf{r} \cdot \mathsf{S}) = \mathsf{r} \cdot \mathsf{P}^{\mathsf{T}} \cdot \mathsf{K}^{\mathbf{memb}} \cdot \mathsf{r} \cdot \mathsf{P}$ and the membrane energy $\mathsf{E}^{\mathbf{memb}}_{\mathsf{S}}(\mathsf{r} \cdot \mathsf{S}) = \mathsf{r}^2 \cdot \mathsf{E}^{\mathbf{memb}}_{\mathsf{S}}(\mathsf{S})$. A similar argument was already employed at the beginning of chapter 4 to explain the poor performance of the simple bending energy functional.

The data dependent bending energy $\mathsf{E}_{\mathfrak{SCh}}(\mathsf{S})$ shows significant errors on regular patches that neighbor extraordinary patches. The reason for this will be analyzed in the next sections.

6.2 Subdivision Scaling of a single Patch

For the experiments in this section three patches of valence 3, 4, 7 and their corresponding one-rings were extracted from the initial cone mesh. An "extraordinary" point was fixed in the valence 4 case and the subdivision operator was applied between $\mathsf{1}$ and $\mathsf{10}$ times. On each subdivision level only the patch next to the extraordinary point was considered in the computations for figure 6.3.

It was attempted to compute the integrals for the exact reference solutions $\mathsf{E}^{\mathbf{memb}}_{\mathsf{S}}(\mathsf{S})$ with $\mathsf{10}$ decimal digits and $\mathsf{E}^{\mathbf{bend}}_{\mathsf{S}}(\mathsf{S})$ with $\mathsf{6}$ decimal digits of accuracy. It is not clear if this was always successfully achieved, especially in the bending energy case. It is notoriously difficult to compute two-dimensional integrals of unbounded integrands without specialized quadrature rules.

We observe, that the relative errors of the data dependent membrane energy $\mathsf{E}^{\mathbf{memb}}_{\mathfrak{SCh}}(\mathsf{S})$ get small very quickly. The relative errors of the simple bending energy $\mathsf{E}^{\mathbf{bend}}_{\mathfrak{Ch}}(\mathsf{S})$ stay big (and on a closer examination we notice, that the errors grow with each subdivision). The relative errors of the simple membrane energy $\mathsf{E}^{\mathbf{memb}}_{\mathfrak{Ch}}(\mathsf{S})$ are reasonably small. For valence $\mathsf{7}$ they even seem to go to $\mathsf{0}$. The relative errors of the data dependent bending energy $\mathsf{E}^{\mathbf{bend}}_{\mathfrak{SCh}}(\mathsf{S})$ are also reasonably small for valences 3 and 7.

The behavior of $\mathsf{E}^{\mathbf{bend}}_{\mathfrak{SCh}}(\mathsf{S})$ for valence 4 needs to be examined in greater detail. Let us recall, that the operator $\mathsf{E}^{\mathbf{bend}}$ is computed by evaluating the Hessian of scalar functions with respect to

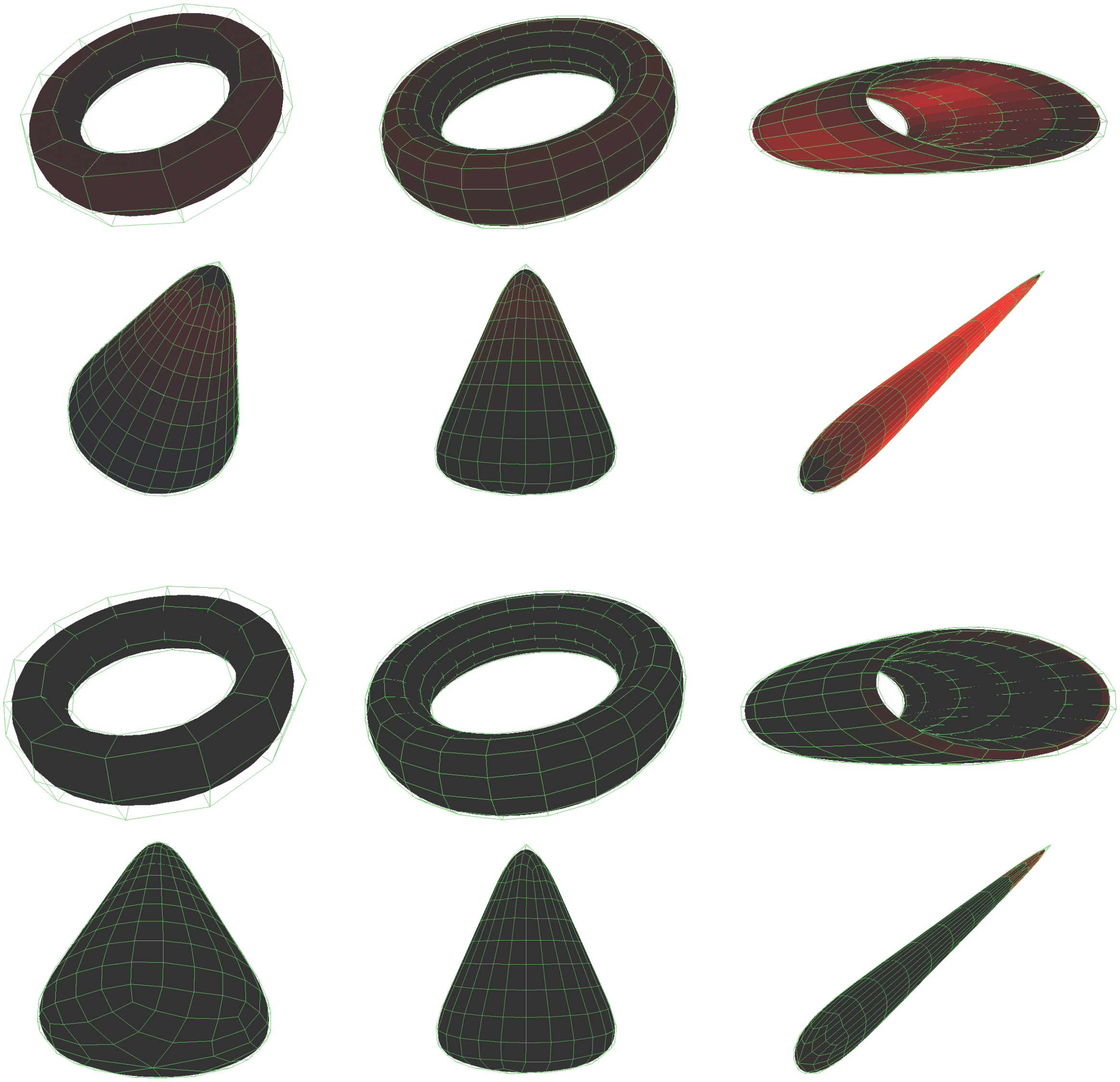


Figure 6.1: Local relative errors of the membrane energy functionals (top two rows simple functional, bottom two rows first order data dependent functional). From left to right the maps I, U and M have been used to transform the control points of the initial mesh. A saturated red color stands for an energy approximation that is at least 4 times too high compared with the exact functional. A saturated blue color symbolizes an energy approximation that is at least by a factor of 4 too low compared with the exact functional. Sizes of images are scaled uniformly to fit the page.

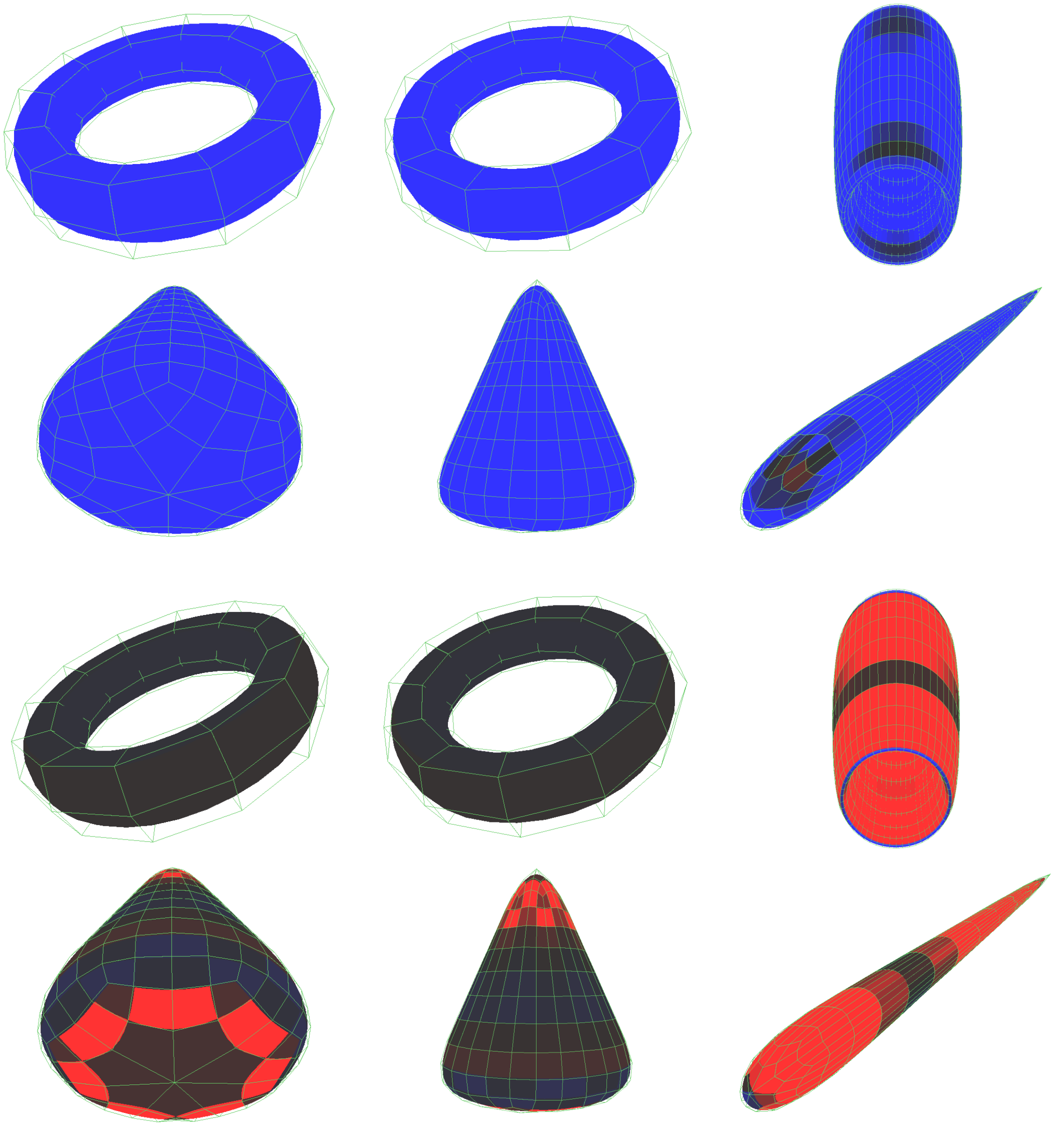


Figure 6.2: Local relative errors of the bending energy functionals (top two rows simple functional, bottom two rows first order data dependent functional). From left to right the maps I, U and M have been used to transform the control points of the initial mesh. A saturated red color stands for an energy approximation that is at least 4 times too high compared with the exact functional. A saturated blue color symbolizes an energy approximation that is at least by a factor of 4 too low compared with the exact functional. Sizes of images are scaled uniformly to fit the page.

their reference surface.

$$\mathbf{Hess}_S(h) = I_S^{-1} \cdot \big(\mathbf{Hess}(h) - (\partial_1 h\Gamma^1 + \partial_2 h\Gamma^2)\big) \tag{6.2}$$

If the reference surface is a regular patch, then the Christoffel symbols of this patch will in all generality be not equal to zero. But for the computation of the first order functionals of regular patches the reference surface $\mathfrak{S}\mathfrak{C}\mathfrak{h}$ chosen is a linear map of the unit square (compare with figure 4.1). This means the Christoffel symbols of $\mathfrak{S}\mathfrak{C}\mathfrak{h}$ are zero for any parameter $\bar{u} \in [0,1]^2$. This “simplifies” the Hessian of h with respect to $\mathfrak{S}\mathfrak{C}\mathfrak{h}$ to

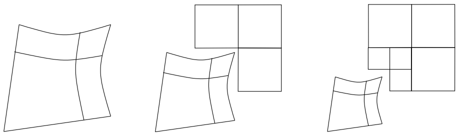
$$\mathbf{Hess}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(h) = I_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}^{-1} \cdot \mathbf{Hess}(h). \tag{6.3}$$

Remember that the parameters of W where selected, such that $I_S \approx I_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}$. This means (at least for valence 4) we can't expect having $E^{\mathbf{bend}}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(S)/E^{\mathbf{bend}}_S(S) \to 1$, if the subdivision level n goes to infinity. For irregular patches of valence other than 4 and arbitrary configuration of the control points it is not as simple to decide the behavior of $E^{\mathbf{bend}}_{\mathfrak{S}\mathfrak{C}\mathfrak{h}}(S)/E^{\mathbf{bend}}_S(S)$. Nevertheless it seems to be clear from the experiments, that all considered energy approximations of a single, repeatedly subdivided extraordinary patch go to zero.

6.3 Subdivision Refinement of Meshes

Subdivision does not change the limit surface of a mesh. But subdivision clearly changes the size and number of patches of the mesh and hence the underlying parametrization.

For regular regions and extraordinary patches the analysis of the previous section carries over to the whole mesh. It does not apply to the neighborhood of the extraordinary patches, where new regular patches are created during the subdivision process.



Let S_n be the refinement of the initial patch S on subdivision level n. Clearly S_n consists of 4^n patches. Let R_n be the corresponding first order data dependent reference surface. Ignoring the linear maps W the reference surface R_n at subdivision level n of the initial extraordinary patch is

$$R_n = \bigcup_{i=0}^{n-1} 2^{-i} \cdot L \ \cup \ \bigcup_{i=n}^{\infty} \lambda^i \cdot L^{\mathfrak{C}\mathfrak{h}}. \tag{6.4}$$

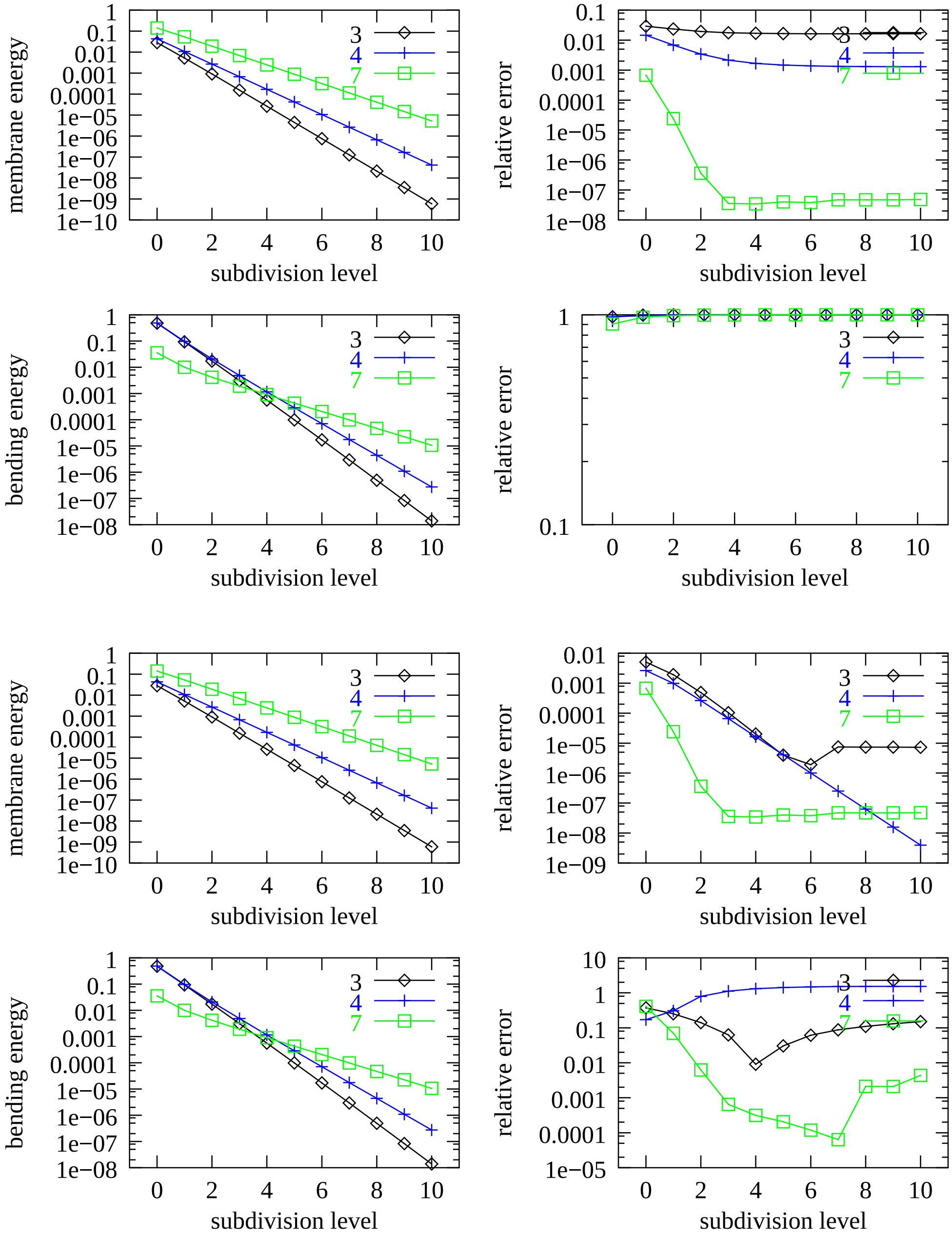


Figure 6.3: Simple energies and their corresponding relative errors (top two rows) and data dependent energies with corresponding relative errors (bottom two rows) of repeatedly subdivided patches.

subdivision level	$E_S^{\mathbf{memb}}(S)$	$E_{\mathfrak{Ch}}^{\mathbf{memb}}(S)$	$E_{\mathfrak{SCh}}^{\mathbf{memb}}(S)$
0	1.213e+01	1.339e+01	1.218e+01
1	1.213e+01	1.341e+01	1.215e+01
2	1.213e+01	1.342e+01	1.214e+01

subdivision level	$E_S^{\mathbf{bend}}(S)$	$E_{\mathfrak{Ch}}^{\mathbf{bend}}(S)$	$E_{\mathfrak{SCh}}^{\mathbf{bend}}(S)$
0	5.720e+01	1.347e+00	1.011e+02
1	5.720e+01	4.626e-01	1.475e+02
2	5.720e+01	1.556e-01	2.019e+02

Figure 6.4: Numerical values for the energy approximations of the whole meshes in figure 6.5

This means increasing the subdivision level n destroys the carefully introduced characteristic map parametrization around the extraordinary vertex and replaces it for $n \to \infty$ with the parametrization used in [HKD93]. For any $n \in \mathbb{N}$ the energy approximation $E_{R_n}^{\mathbf{bend}}(S)$ is finite. Still for many initial configurations of the control points $E_{R_n}^{\mathbf{bend}}(S) \to \infty$ for $n \to \infty$.

This property is clearly disappointing. It could be avoided, if we only had for regular patches $E_{\mathfrak{SCh}_n}^{\mathbf{bend}}(S)/E_S^{\mathbf{bend}}(S) \to 1$ with a sufficiently high convergence rate! This is not true for our functionals $E_{\mathfrak{SCh}}^{\mathbf{bend}}(S)$, as we have already seen in the previous section. The good news is, that previously developed methods for bicubic B-spline meshes (similar to [MS92] and [GLW96]) provide us with a functional for which at least on bicubic B-spline meshes $E_{R_n}^{\mathbf{bend}}(S)/E_S^{\mathbf{bend}}(S) \to 1$. It still needs to be shown that the convergence rate can be chosen high enough such that hopefully $E_{R_n}^{\mathbf{bend}}(S) \to S_S^{\mathbf{bend}}(S)$ for $n \to \infty$ for subdivision surfaces S.

Figures 6.4 and 6.5 illustrate the behavior of the data dependent energy functionals $E_{\mathfrak{SCh}}^{\mathbf{memb}}(S)$ and $E_{\mathfrak{SCh}}^{\mathbf{bend}}(S)$ for different levels of subdivision. The data for $E_{\mathfrak{Ch}}^{\mathbf{bend}}(S)$ in the second table of figure 6.4 underscores its generally undesirable behavior.

6.4 Derivatives of the Energy Functionals

The simple and data dependent gradients of the bending energy of a subdivided cube are shown in figure 6.6. We observe, that both gradients are very long compared with the dimensions of the model. This is a commonly encountered situation. We also note, that the gradients differ in length and direction.

Figure 6.7 clearly illustrates, that there is no interaction between the x, y and z components of the control points in the computation of the simple bending energy. As a consequence the equation system used in [HKD93] to fair the Catmull-Clark surface decoupled into three smaller systems. The

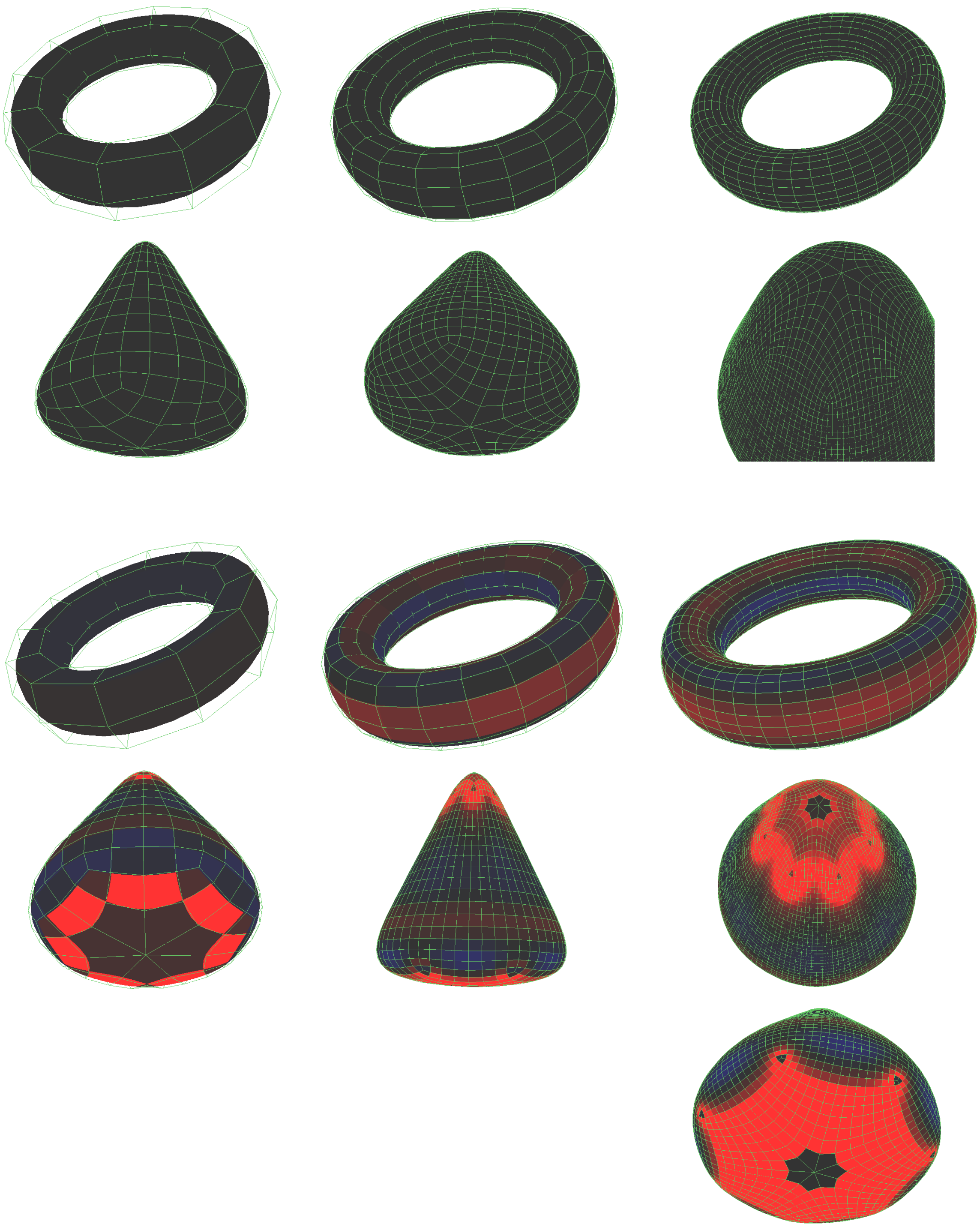


Figure 6.5: Local relative errors of the first order data dependent membrane (top two rows) and first order data dependent bending energy functionals (bottom rows) shown for three levels of subdivision. The cone is shown twice from top and bottom for the bending energy on the third level of subdivision (right hand side). A saturated red color stands for an energy approximation that is at least 4 times too high compared with the exact functional. A saturated blue color symbolizes an energy approximation that is at least by a factor of 4 too low compared with the exact functional. Sizes of images are scaled uniformly to fit the page.

three blocks on the diagonal are equal, simplifying the process even further.

In contrast to that the Hessian of the first order data dependent energy has nonzero entries outside of the block-diagonal matrices. All 9 blocks have the same sparsity structure, but in general none of them are equal. The interaction between the x, y and z components of the control points might be seen as a hint, that the data dependent functionals could be able to capture qualitatively different effects than the simple functional.

The Hessians of the simple functionals are always positive (semi-)definite. It has been experimentally verified, that this is not always the case for the Hessians of the first order data dependent functionals.

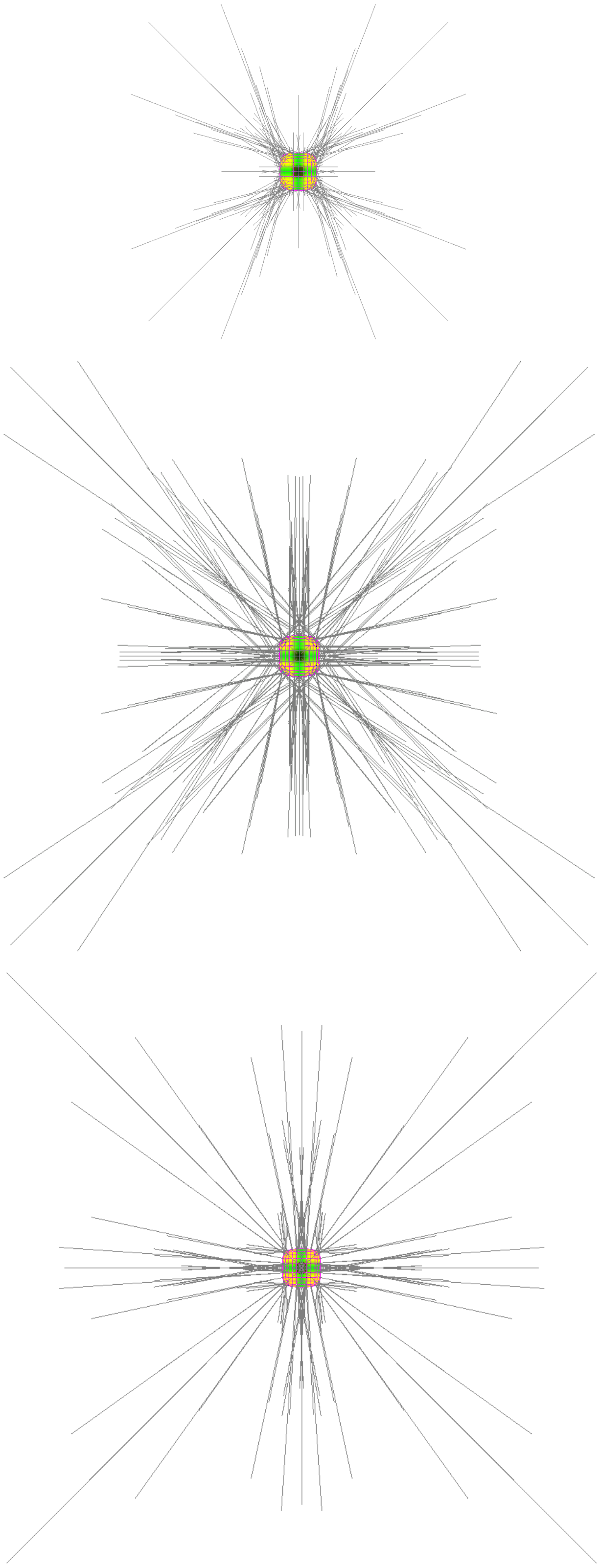


Figure 6.6: Gradients of the bending energy of the cube. The top image shows the gradient of the simple, the center image the gradient of the first order data dependent and the bottom image the gradient of the exact bending energy functional. The colors of the cube show the mean curvature.

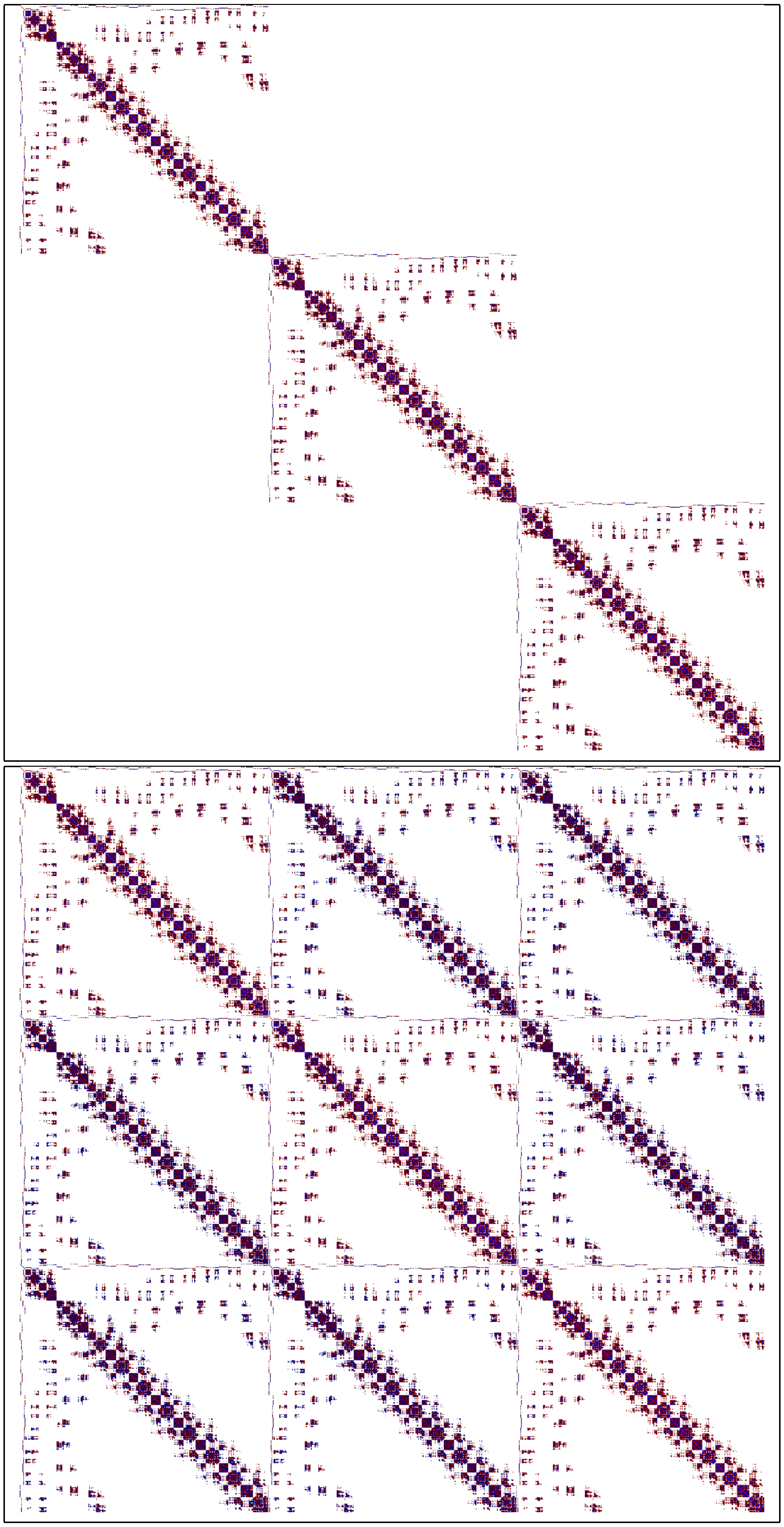


Figure 6.7: The Hessians of the bending energy of the cube. The simple functional (top) has a block-diagonal structure. The data dependent functional (bottom) not. Both Hessians are sparse because of the local support of the basis functions of the Catmull-Clark subdivision scheme. Red color denotes positive entries, negative entries are colored blue.

Chapter 7

Conclusion

In chapter 3 we computed the simple energy functionals for extraordinary patches avoiding divergent integrals using the characteristic map parametrization. Based on this result we introduced in chapter 4 new data dependent functionals. It was shown how to evaluate these functionals quickly by precomputing 3 respectively 6 matrices and scaling them when needed with simple factors. In chapter 5 we showed, that it is possible to compute with high accuracy the gradient and Hessian of the new energies. These computations are efficient with respect to memory overhead and evaluation time.

In chapter 6 the simple and the data dependent functionals where compared to their corresponding exact parametrization independent solutions. It was shown that the first order data dependent membrane energy behaves nearly as well as the parametrization independent energy. It was argued, that the simple bending energy is not a good model in many cases. The simple membrane and the first order data dependent bending energy where shown to be acceptable models for their corresponding exact energies in simple situations only.

It is desirable to analyse the behavior of the first order data dependent energies further. The experimental analysis is currently slowed down by the computation of the exact functionals. This could be changed by implementing specialized integration rules. In case the first order data dependent functionals prove themselves in applications, it would be necessary to obtain formal proofs of some of the experimentally shown approximation and convergence properties.

An important question that remains to be examined, is if and how one should try to fix the blow-up of the first order data dependent bending energy error next to extraordinary patches. The author hopes it could be done with methods previously developed for bicubic B-splines.

To evaluate the performance of the new operators in a real world application the author plans to implement a non-linear high dimensional optimizer for the fairing of subdivision surfaces.

Appendix A

Used Symbols

Symbol	Domain	
S	$\mathbb{R}^2 \rightarrow \mathbb{R}^3$	surface or patch
R	$\mathbb{R}^2 \rightarrow \mathbb{R}^3$	reference surface or patch
$E_R(S)$		energy of patch S with respect to reference surface R
$P, \bar{X}$	$\mathbb{R}^{3M}$	control points
K	$\mathbb{R}^{M \cdot M}$	quadratic form
ϕ_i	$\mathbb{R}^2 \rightarrow \mathbb{R}$	eigen basis function of patch
λ_i	$\mathbb{R}$	eigen value belonging to ϕ_i, $\lambda_i \geq \lambda_j$ if $i > j$
$\mathbf{I}_S$	$\mathbb{R}^2 \rightarrow Mat(2,2)$	first fundamental form of S at
$\mathbf{II}_S$	$\mathbb{R}^2 \rightarrow Mat(2,2)$	second fundamental form of S at
Γ_{ij}^k	$\mathbb{R}^2 \rightarrow \mathbb{R}$	Christoffel symbols
$\mathbf{grad}_R(h)$		gradient of h with respect to reference surface R
$\mathbf{Hess}_R(h)$		Hessian of h with respect to reference surface R

Bibliography

[GKS02] Eitan Grinspun, Petr Krysl, and Peter Schröder. CHARMS: A simple framework for adaptive simulation. *To appear in Proceedings of SIGGRAPH*, 2002.

[GLW96] G. Greiner, J. Loos, and W. Wesselink. Data dependent thin plate energy and its use in interactive surface modeling. In *Computer Graphics Forum (Proc. EUROGRAPHICS '96), 15(3)*, pages 175–186, 1996.

[HKD93] Mark Halstead, Michael Kass, and Tony DeRose. Efficient, fair interpolation using Catmull-Clark surfaces. *Proceedings of SIGGRAPH 93*, pages 35–44, 1993.

[MS92] H.P. Moreton and C.H. Sequin. Functional optimization for fair surface design. *Computer Graphics*, 26(Annual Conference Series):167–176, 1992.

[Nit75] J. C. C. Nitsche. *Vorlesungen über Minimalflächen.* Springer-Verlag, 1975.

[NW99] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer, 1999.

[Rei95] Ulrich Reif. A unified approach to subdivision algorithms near extraordinary vertices. *Computer Aided Geometric Design*, 12(2):153–174, 1995.

[Rei98] Ulrich Reif. *Analyse und Konstruktion von Subdivisionsalgorithmen für Freiformflächen beliebiger Topologie.* Department of Mathematics, Stuttgart University, 1998.

[RS01] Ulrich Reif and Peter Schröder. Curvature integrability of subdivision surfaces. *Advances in Computational Mathematics*, 2, 2001.

[Sta98] Jos Stam. Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. *Computer Graphics*, 32(Annual Conference Series):395–404, 1998.

[Tho79] John A. Thorpe. *Elementary topics in differential geometry.* Springer, 1979.

[Tip98] Colin Tipping. Technical change and the ship draughtsman. *The Mariner's Mirror*, 84:458–469, 1998.