

Discrete Fairing

Leif Kobbelt

University of Erlangen-Nürnberg

Abstract

We address the general problem of, given a triangular net of arbitrary topology in \mathbb{R}^3 , find a refined net which contains the original vertices and yields an improved approximation of a smooth and fair interpolating surface. The (topological) mesh refinement is performed by uniform subdivision of the original triangles while the (geometric) position of the newly inserted vertices is determined by variational methods, i.e., by the minimization of a functional measuring a discrete approximation of bending energy. The major problem in this approach is to find an appropriate parameterization for the refined net's vertices such that second divided differences (derivatives) tightly approximate intrinsic curvatures. We prove the existence of a unique optimal solution for the minimization of discrete functionals that involve squared second order derivatives. Finally, we address the efficient computation of fair nets.

1 Introduction

One of the main problems in geometric modeling is the generation of aesthetically appealing surfaces. Usually these surfaces are subject to technical requirements like interpolation constraints. While the constraints can easily be formulated in mathematical terms (and thus are compatible to the mathematical description of the surface itself), the explicit formulation of 'well-shaped-ness' causes some difficulties. Motivated by physical models of elastic membranes or thin plates, the variational approach to surface design measures the 'bad-shaped-ness' of a surface by the value of some (bending-) energy functional. Surfaces of minimal energy are assumed to be *fair* [Sap94].

Suitable energy functionals are defined in terms of geometric invariants like principal curvatures and fundamental forms. While such concepts are well established in differential geometry, they do not fit easily into the framework of piecewise polynomial parametric surfaces which is how surfaces are usually represented in CAGD.

The main source for these problems is that for any basis representation of a polynomial patch the intrinsic geometric measures depend on the coefficients in a non-linear fashion and their exact evaluation turns out to be rather complex [MS92]. Hence, the common approach to fairing schemes is to find a parameterization which allows to approximate geometric curvatures simply by second order derivatives (which are linear in the coefficients).

Once an eligible parameterization is found, we can exploit results from variational calculus [Els70] and Galerkin-projections reduce the optimization to the solution of a sparse linear system. Thus, much of the work in this field addresses various methods to find close-to-isometric parameterizations for surfaces of arbitrary shape.

Under the assumption that a parametric surface is rather flat, the metric on the surface does not differ very much from the metric in the plane. Therefore, geometric curvature measures like $(\kappa_1 + \kappa_2)^2$, $|\kappa_1 \kappa_2|$, or $\kappa_1^2 + \kappa_2^2$ can be approximated by combinations of second order derivatives.

Although this typically does *not* hold for realistic surface modeling tasks, it is usually assumed, e.g., in [CG91], [HKD93], and the parametric situation is treated exactly like

the (piecewise) functional case [Duc79], [Pow94]. More sophisticated linearizations of the fairness functional are used in [Gre94], [GLW96] where a non-flat reference surface provides the parameter domain instead of the plane.

A completely different approach to surface generation are *subdivision techniques*. Interpolatory refinement schemes like [DGL90], [ZSS96], [Kob96b] map a given triangular or quadrilateral net to a refined net by inserting new vertices whose location is computed by a linear combination of vertices from the given net. By iterating this subdivision step, a sequence of meshes is generated which converges to a smooth surface.

In [Kob96a] the concepts of subdivision and fairing are combined in order to define univariate *variational refinement schemes*. Such schemes use the refinement paradigm to generate sequences of polygons while the position of the new vertices is determined by minimizing an appropriate energy functional.

In the present paper, a similar approach is followed in the bivariate setting of (iteratively) refined triangular meshes. Given an initial net whose vertices are to be interpolated, we generate a topologically refined mesh by uniform subdivision of the faces. The position of the newly inserted vertices is found by solving an optimization problem which discretizes a variational problem.

We generalize variational subdivision as presented in [Kob96a] by no longer restricting ourselves to *binary* subdivision. However, when it comes to the efficient implementation of the proposed scheme, we have to adapt multi-grid techniques to solve the occurring linear systems. For this we need a sequence of nested spaces which are most easily defined by iterative binary subdivision. Hence, for implementation purposes we will return to binary subdivision with the important difference that, in order to minimize the objective energy functional, *all* vertices but the original ones are allowed to move while in [Kob96a] in each step only the newest vertices move and all intermediate ones are kept fixed.

2 Uniform subdivision of triangular nets

We use uniform subdivision to generate triangular nets on which the fairing is performed. We distinguish between topological mesh refinement and geometric net smoothing. The smoothing will be achieved by energy minimization.

Let $N := (\mathcal{P}, \mathcal{T})$ be a triangular net with vertices $\mathbf{p}_i \in \mathcal{P} \subset \mathbb{R}^3$ and faces $[\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k] = \{\alpha \mathbf{p}_i + \beta \mathbf{p}_j + \gamma \mathbf{p}_k, \alpha, \beta, \gamma \geq 0, \alpha + \beta + \gamma = 1\} \in \mathcal{T}$. Each edge $\overline{\mathbf{p}_i \mathbf{p}_j}$ may be part of at most two adjacent triangles. If it belongs to only one triangle, the edge is a *boundary edge*. To avoid numerical instabilities and the treatment of special cases, we assume that the inner angles of triangles in the original net N are bounded below by $\alpha_\varepsilon > 0$.

A *uniform refinement operator* \mathbf{S}_r ($r \geq 2$) maps a given net N to $\mathbf{S}_r N = (\mathcal{P}', \mathcal{T}')$ with

$$[\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k] \mapsto \left\{ \begin{array}{c} \{[\mathbf{q}_{u+1,v,w}, \mathbf{q}_{u,v+1,w}, \mathbf{q}_{u,v,w+1}], 0 \leq u, v, w \in \mathbb{N}, u+v+w = r-1\} \\ \cup \\ \{[\mathbf{q}_{u-1,v,w}, \mathbf{q}_{u,v-1,w}, \mathbf{q}_{u,v,w-1}], 1 \leq u, v, w \in \mathbb{N}, u+v+w = r+1\} \end{array} \right\}.$$

For simplicity we omit the additional indices i, j and k in $\mathbf{q}_{i,j,k,u,v,w}$ if it is obvious or irrelevant to which original triangle a new vertex belongs. The refinement is called *interpolatory* if $\mathcal{P} \subset \mathcal{P}'$ and $\mathbf{q}_{r,0,0} = \mathbf{p}_i$, $\mathbf{q}_{0,r,0} = \mathbf{p}_j$, and $\mathbf{q}_{0,0,r} = \mathbf{p}_k$.

For $r = 2$ the subdivision operator \mathbf{S}_2 is called *binary*. Iterating a binary subdivision operator, leads to refined nets $(\mathbf{S}_2)^n N = \mathbf{S}_{2^n} N$. Hence, the more general definition of *r-nary* subdivision allows a more flexible adaption of the resolution.

The most simple subdivision operator \mathbf{L}_r reproduces piecewise linear surfaces. It computes the position of the new vertices by

$$\mathbf{q}_{u,v,w} := \frac{1}{r} (u \mathbf{p}_i + v \mathbf{p}_j + w \mathbf{p}_k). \quad (2.1)$$

We classify the vertices $\mathbf{q}_{u,v,w} \in \mathcal{P}'$ into *inner vertices* with $u, v, w \geq 1$, *edge vertices* with exactly one barycentric index vanishing, and *corner vertices*: $\mathbf{q}_{r,0,0}, \mathbf{q}_{0,r,0}, \mathbf{q}_{0,0,r}$. The corner vertices unambiguously correspond to the vertices of the unrefined net N (for interpolatory refinement they coincide) and the edge vertices are topologically associated with the edges of N . Since all inner and edge vertices have valence 6, the topological structure of the refined net $\mathbf{S}_r N$ is characterized by isolated singular vertices (among the corner vertices) being separated by regular subnets (*subdivision topology*).

3 Fairing triangular nets

The term *fairing* usually refers to the definition of (continuous) surfaces by variational methods, i.e., by the solution of an optimization problem. To improve the quality of free form surfaces satisfying a set of interpolation conditions, one usually fixes the undetermined degrees of freedom by minimizing an appropriate fairing functional.

We start with the functional setting and generalize to parametric surfaces later. The total strain energy of a function $f : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ with $f \in X \subset C^k(\Omega)$ for a sufficiently large k , can be measured by a semi-norm $\|f\|_D^2 := \langle Df, Df \rangle$ induced by an appropriately chosen inner product

$$\langle Df, Dg \rangle := \int_{\Omega} \sum_{i=0}^{r-1} (D_i f) (D_i g), \quad (3.1)$$

where $D = [D_i]_{i=0}^{r-1} := \text{diag}[\rho_0, \dots, \rho_{r-1}] \mathcal{D}_k$ represents a vector of differential operators, $\text{diag}[\rho_i]$ is the diagonal matrix with elements ρ_i , and \mathcal{D}_k is the k -jet operator. The graph of a function f minimizing $\|f\|_D^2$ is called a *fair surface*. Due to the geometric background of the problem, rotationally invariant functionals are particularly interesting. Those can be obtained by using linear combinations of

$$\left[\underbrace{0, \dots, 0}_{\frac{1}{2}(k+1)k}, \dots, \underbrace{\sqrt{\binom{k}{i}} \frac{\partial^k}{\partial u^{k-i} \partial v^i}, \dots}_{i=0, \dots, k} \right]^T, \quad k \in \mathbb{N}. \quad (3.2)$$

To control the shape of the surface, in addition to optimality some interpolation conditions can be imposed. When interpolating discrete positional (scattered) data $\mathbf{p}_i := (x_i, y_i, f_i) = (x_i, y_i, f(x_i, y_i))$ for $i = 1, \dots, n$, we can satisfy these conditions by representing the solution $f = f^* - \tilde{f}$ by an arbitrary (fixed) interpolant $f^* \in X$ plus $\tilde{f} \in \ker(V) := \ker[\delta_{(x_i, y_i)}]_{i=1}^n \subset X$. The operator $V := [\delta_{(x_i, y_i)}]_{i=1}^n$ is called the *data functional*; $\delta_{(x_i, y_i)}(f) = f(x_i, y_i)$ is the *point evaluation*.

Assume that $f \in X$ minimizes $\|f\|_D^2 = \langle Df, Df \rangle$ and consider

$$s(\lambda) := \langle D(f + \lambda g), D(f + \lambda g) \rangle$$

with $g \in \ker(V)$ arbitrary. Since $\|f\|_D^2$ is a minimum, the function s has a minimum at $\lambda = 0$, i.e.,

$$\frac{d}{d\lambda} s(\lambda) \Big|_{\lambda=0} = 2 \langle Df, Dg \rangle + 2\lambda \langle Dg, Dg \rangle \Big|_{\lambda=0} = 2 \langle Df, Dg \rangle \stackrel{!}{=} 0.$$

Hence

$$\langle Df, Dg \rangle = 0, \quad \forall g \in \ker(V) \quad (3.3)$$

characterizes the solution f with minimum energy $\|f\|_D^2$.

Using *Galerkin*-projections, the space X is defined by a set of basis functions, i.e., $X := \text{span}\{\phi_1, \dots, \phi_m\}$ with $m \geq n$. For $f := \Phi([\alpha_j]) = \sum_j \alpha_j \phi_j$ and $g := \Phi([\beta_l])$, equation (3.3) can be rewritten as

$$[\beta_l]^T [\langle D \phi_l, D \phi_j \rangle] [\alpha_j] = 0$$

since D is a linear operator. Let $B : \mathbb{R}^d \rightarrow \mathbb{R}^m$ be a basis of $\ker(V \Phi)$. Equation (3.3) holding for all $g \in \ker(V)$ is equivalent to $B^T [\langle D \phi_l, D \phi_j \rangle] [\alpha_j]$ being the zero matrix. All vectors $[\alpha_j]$ for which $\Phi([\alpha_j])$ satisfies the interpolation conditions, have the form $[\alpha_j] = \mathbf{a}^* - B \tilde{\mathbf{a}}$ with $\Phi(\mathbf{a}^*)$ being any interpolant and $\tilde{\mathbf{a}} \in \mathbb{R}^d$ (hence $\Phi(B \tilde{\mathbf{a}}) \in \ker(V)$). A solution of

$$B^T [\langle D \phi_l, D \phi_j \rangle] B \tilde{\mathbf{a}} = B^T [\langle D \phi_l, D \phi_j \rangle] \mathbf{a}^* \quad (3.4)$$

therefore provides a minimizing function $f := \Phi(\mathbf{a}^* - B \tilde{\mathbf{a}}) \in \mathbf{X}$. The existence of a unique solution $\tilde{\mathbf{a}}$ of (3.4) follows if $\ker(B^T [\langle D \phi_l, D \phi_j \rangle] B) = \{0\}$ or, equivalently

$$\ker \langle D \cdot, D \cdot \rangle \cap \text{ran}(\Phi B) = \ker \langle D \cdot, D \cdot \rangle \cap \ker(V) = \{0\}. \quad (3.5)$$

The equivalence can be seen as follows: $\Phi(B \tilde{\mathbf{a}})$ lies in $\ker \langle D \cdot, D \cdot \rangle$ if and only if $D \Phi(B \tilde{\mathbf{a}}) \equiv 0$ since $\langle \cdot, \cdot \rangle$ is a norm on X . But then $\tilde{\mathbf{a}} \in \ker([\langle D \phi_l, D \phi_j \rangle] B) \subset \ker(B^T [\langle D \phi_l, D \phi_j \rangle] B)$. On the other hand $\tilde{\mathbf{a}} \in \ker(B^T [\langle D \phi_l, D \phi_j \rangle] B)$ implies $\tilde{\mathbf{a}}^T B^T [\langle D \phi_l, D \phi_j \rangle] B \tilde{\mathbf{a}} = 0$ and hence $\Phi(B \tilde{\mathbf{a}}) \in \ker \langle D \cdot, D \cdot \rangle$. \square

The restriction to diagonal transformations of \mathcal{D}_k in the definition of D makes it easy to determine the kernel of the semi-norm $\|\cdot\|_D^2 = \langle D \cdot, D \cdot \rangle$. For the examples in (3.2) the kernel consists of all polynomials with maximum total degree less than k . Hence, if the abscissae of the data points $\mathbf{p}_i|_{x,y}$ do not lie on an algebraic curve of degree less than k , the minimization has a unique solution¹.

3.1 Discretization \rightarrow Difference methods

Instead of restricting to the finite dimensional space $X = \text{span}\{\phi_1, \dots, \phi_m\}$, we can set $X = C^k(\Omega)$ and approximate (3.3) by discretization of the domain Ω over a finite point set $\mathcal{Q} = \{(x_l, y_l)\} \subset \Omega$ with $\{\mathbf{p}_i|_{x,y}\} \subset \mathcal{Q}$ (interpolation). We want to compute the function values f_l at $(x_l, y_l) \in \mathcal{Q} \setminus \{\mathbf{p}_i|_{x,y}\}$ corresponding to a fair surface.

First, we replace in (3.1) the integral over Ω by a quadrature formula with positive weights w_j

$$\langle D f, D g \rangle \approx \sum_j w_j \sum_{i=0}^{r-1} (D_i f)(\xi_j, \eta_j) (D_i g)(\xi_j, \eta_j). \quad (3.6)$$

Then the differential operators D_i evaluated at the points (ξ_j, η_j) are replaced by divided difference operators with masks $[\gamma_{i+rj,l}]_l$

$$\langle D f, D g \rangle \approx \sum_j w_j \sum_{i=0}^{r-1} \left(\sum_l \gamma_{i+rj,l} f_l \right) \left(\sum_l \gamma_{i+rj,l} g_l \right) = [f_l]^T \Gamma^T \Gamma [g_l], \quad (3.7)$$

where $\Gamma := [\sqrt{w_j} \text{div}_r \gamma_{j,l}]_{j,l}$. The coefficients $\gamma_{j,l}$ can be chosen such that they vanish outside some neighborhood $\mathcal{U}_j \subset \mathcal{Q}$ of $(\xi_j \text{div}_r, \eta_j \text{div}_r)$ which makes Γ a sparse matrix. Let the vector $[f_l]$ be decomposed into $[f_l] = [\tilde{f}_l; \hat{f}_l]$ where the components \tilde{f}_l correspond to

¹All through the paper we use bold letters \mathbf{p} for vectors of any dimension and the projection operators $\mathbf{p}|_u$ and $\mathbf{p}|_{u,v} = (\mathbf{p}|_u, \mathbf{p}|_v)$ to select specific coordinates.

the locations $\mathbf{p}_i|_{x,y}$ and are therefore fixed by the interpolation conditions and \tilde{f}_i are the free variables. We decompose $[g_i]$ and Γ accordingly

$$[\bar{f}_i; \tilde{f}_i]^T \Gamma^T \Gamma [\bar{g}_i; \tilde{g}_i] = [\bar{f}_i; \tilde{f}_i]^T \begin{pmatrix} \bar{\Gamma}^T \\ \tilde{\Gamma}^T \end{pmatrix} (\bar{\Gamma}, \tilde{\Gamma}) [\bar{g}_i; \tilde{g}_i] = [\bar{f}_i; \tilde{f}_i]^T \begin{pmatrix} \bar{\Gamma}^T \bar{\Gamma} & \bar{\Gamma}^T \tilde{\Gamma} \\ \tilde{\Gamma}^T \bar{\Gamma} & \tilde{\Gamma}^T \tilde{\Gamma} \end{pmatrix} [\bar{g}_i; \tilde{g}_i].$$

Since $g \in \ker(V)$, we have $[\bar{g}_i] = 0$ and hence

$$[\bar{f}_i; \tilde{f}_i]^T \Gamma^T \Gamma [\bar{g}_i; \tilde{g}_i] = ([\bar{f}_i]^T \bar{\Gamma}^T \tilde{\Gamma} + [\tilde{f}_i]^T \tilde{\Gamma}^T \bar{\Gamma}) [\tilde{g}_i],$$

which has to be zero for all $[\tilde{g}_i]$ if (3.7) is considered to be a discrete approximation of (3.3). The function values $[\tilde{f}_i]$ corresponding to a fair surface are thus given by a solution of

$$\tilde{\Gamma}^T \tilde{\Gamma} [\tilde{f}_i] = -\tilde{\Gamma}^T \bar{\Gamma} [\bar{f}_i]. \quad (3.8)$$

The uniqueness of the solution is determined by the regularity of $\tilde{\Gamma}^T \tilde{\Gamma}$, i.e., by $\ker(\tilde{\Gamma}) = \{0\}$. Since $\|\Gamma [0; \tilde{g}_i]\|_2^2 = [0; \tilde{g}_i]^T \Gamma^T \Gamma [0; \tilde{g}_i] = [\tilde{g}_i]^T \tilde{\Gamma}^T \tilde{\Gamma} [\tilde{g}_i] = \|\tilde{\Gamma} [\tilde{g}_i]\|_2^2$, this is equivalent to

$$\ker(\cdot \Gamma^T \Gamma \cdot) \cap \{[0; \tilde{g}_i]\} = \{0\} \quad (3.9)$$

which turns out to be a discretization of (3.5).

3.2 Fairing uniformly refined nets

Let \mathcal{T} be a triangulation of Ω using the nodes $(x_i, y_i) = \mathbf{p}_i|_{x,y}$ from the given data. By inverse projection $(x_i, y_i) \mapsto \mathbf{p}_i$ we obtain a triangular net $N = (\{\mathbf{p}_i\}, \mathcal{T})$ which serves as input for the discrete optimization. An interpolatory refinement operator \mathbf{S}_r generates a new net $N' = \mathbf{S}_r N$ where the x - and y -coordinates of the new vertices are given by

$$\mathbf{q}_{u,v,w}|_{x,y} := \frac{1}{r} (u \mathbf{p}_i + v \mathbf{p}_j + w \mathbf{p}_k)|_{x,y}, \quad (3.10)$$

i.e., the parameter plane is subdivided piecewise uniformly. Any other subdivision scheme that guarantees to keep a lower bound α_ε for the minimum angle of the triangles in the parameter plane could be used as well. The function values $\mathbf{q}_{u,v,w}|_f$ are found by solving (3.8).

For the construction of the divided difference operators Γ_j up to the order of k at a point (ξ_j, η_j) , we restrict the non-vanishing coefficients γ_l to a neighborhood $\mathcal{U}_j := \{(x_l, y_l)\} \subset \mathcal{Q} := \{\mathbf{q}_{u,v,w}|_{x,y}\} \cup \{\mathbf{p}_i|_{x,y}\}$ with $\#\mathcal{U}_j \geq r := (k+2)(k+1)/2$. We have to estimate the coefficients of a Taylor-expansion at (ξ_j, η_j) . Equivalently, we compute the coefficients of an interpolating polynomial.

Let $W_j := [\frac{1}{a!b!} (x_l - \xi_j)^a (y_l - \eta_j)^b]_{a+b \leq k}$ be the corresponding Vandermonde matrix with the index l being constant for each column. If W_j has the full rank r then numerically stable masks for the difference operators are obtained by the least norm solution

$$\Gamma_j := (W_j W_j^T)^{-1} W_j \quad (3.11)$$

where the i th row of Γ_j contains the masks for the i th difference operator according to the numbering of the rows in W_j . To guarantee numerical stability in this computation it is necessary to introduce the lower bound α_ε for the angles of the triangles in the parameter plane. The matrix Γ_j is a discrete approximation of the k -jet operator at (ξ_j, η_j) and $\text{diag}[\rho_0, \dots, \rho_{r-1}] \Gamma_j$ for all j are the building blocks of the discrete energy functional $\|\Gamma \cdot\|_2^2$.

For discrete fairing, one usually chooses $\#\mathcal{U}_j > r$ in order to generate masks that reflect the rotational invariance of the corresponding continuous functional, i.e., one is usually

interested in symmetric neighborhoods \mathcal{U}_j . One problem that arises from this redundancy is that the kernel of Γ_j becomes bigger with increasing $\#\mathcal{U}_j$ and therefore threatens the solvability condition (3.9). A proof of (3.9) for a particular functional thus has to exploit the correlation between different blocks Γ_j and Γ_l with overlapping domains \mathcal{U}_j and \mathcal{U}_l (cf. Sect. 5).

On the other hand, the sizes of the neighborhoods \mathcal{U}_j determine the degree of sparseness of the matrix $\tilde{\Gamma}^T \tilde{\Gamma}$ in (3.8) and therefore have some impact on the computational complexity of the solution. For the particular difference operators in (3.2) the following (symmetric) neighborhoods are a good compromise between accuracy and efficiency: For even $k = 2k'$ the k' -discs $\mathcal{U} := d_{k'}(\mathbf{p})$ consisting of all vertices having a topological distance to \mathbf{p} no bigger than k' , and for odd $k = 2k' + 1$, the neighborhoods $\mathcal{U} := d_{k'}(\mathbf{p}_i) \cup d_{k'}(\mathbf{p}_j) \cup d_{k'}(\mathbf{p}_l)$ with $[\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_l] \in \mathcal{T}$.

3.3 Generalization to the parametric case

Generalizing the fairing approach from the functional settings to the case of parametric surfaces, i.e., to surfaces which are (piecewisely) defined by functions $\mathbf{f} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$, is not straightforward. One reason for the occurring problems is that the interpolation data $\mathbf{p}_i = (x_i, y_i, z_i)$ for $i = 1, \dots, n$ in this case usually comes without parameter values. Hence, since no particular parameterization of the resulting surface is given naturally (as it is in the functional case), the minimization of derivatives does not have an obvious geometric interpretation. Further, even if we define fairness relative to a special parameterization, we have the difficulty that a regular global parameterization can be defined only for surfaces isomorphic to parts of the torus.

Suppose we had given a fixed parameterization $\mathbf{f} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$. The parametric formulation of the energy functional would be identical to (3.1) with only the simple multiplications of the components $(D_i f)^2$ replaced by dot-products $(D_i \mathbf{f})(D_i \mathbf{f})^T$ according to the differentiation of vector-valued functions

$$D_i \mathbf{f} := (D_i \mathbf{f}|_x, D_i \mathbf{f}|_y, D_i \mathbf{f}|_z).$$

Since the functional (3.1) is quadratic, we can minimize the three components independently from each other:

$$\min_{\mathbf{f}} \int_{\Omega} \sum_{i=0}^{r-1} (D_i \mathbf{f})(D_i \mathbf{f})^T = \sum_{\xi \in \{x,y,z\}} \min_{\mathbf{f}|_{\xi}} \int_{\Omega} \sum_{i=0}^{r-1} (D_i \mathbf{f}|_{\xi})^2. \quad (3.12)$$

This reduces the optimization in the parametric case to three separate functional optimizations. Notice that the three corresponding equations which have to be solved (cf. (3.4) or (3.8) resp.) differ only on their right hand sides.

The graph of a function $f : \Omega \in \mathbb{R}^2 \rightarrow \mathbb{R}$ is a special case of a parametric surface with two coordinate functions being the identity. In the functional case, we used a suitable subdivision scheme (3.10) to compute the x - and y -coordinates of the new vertices directly. In the parametric framework, this can be considered as choosing a piecewise uniform parameterization and assigning the x - and y -values to the new vertices according the (known) optimal coordinate functions (which are both the identity). Obviously, for fairness functionals based on derivatives of second or higher order, the identity is a minimizing function and therefore, the refinement rule (3.10) is not artificial in these cases. It is exactly the solution otherwise obtained by solving the parametric optimization (with respect to this special parameterization).

3.4 Discretization in the parametric case

For the discretization in the functional case (Sect. 3.1) we had to assign parameter values to vertices in the neighborhoods $\mathcal{U}_j = \mathcal{U}(\xi_j, \eta_j)$ in order to construct divided difference operators. The discretization in the parametric case now goes along the same lines, just that the parameter values are not implicitly given by the input data. Further, since in general for parametric surfaces no regular global parameterization exists, we have to generalize the setting by allowing vertices to have multiple parameter values assigned.

Let $\{\mathcal{U}_1, \dots, \mathcal{U}_m\}$ be a collection of local neighborhoods covering the refined net $N' = \mathbf{S}_r N = (\mathcal{P}', \mathcal{T}')$ with $\mathcal{U}_j \subset \mathcal{P}'$, $\#\mathcal{U}_j \geq r := \frac{1}{2}(k+2)(k+1)$ for all j , and $\bigcup \mathcal{U}_j = \mathcal{P}'$. A *local parameterization (a chart)* is a map $\mu_j : \mathcal{U}_j \rightarrow \mathbb{R}^2$ which assigns parameter values to all vertices in the set \mathcal{U}_j . Since every vertex may belong to several neighborhoods, different parameter values may be assigned to it in each of them. In order to construct k -th order difference operators on \mathcal{U}_j , the set $\mu_j(\mathcal{U}_j)$ has to be such that there exists no non-trivial polynomial of degree k which vanishes at all $(u_l, v_l) \in \mu_j(\mathcal{U}_j)$. If this condition is satisfied, the Vandermonde-matrices $W_j = [\frac{1}{a!b!} u_l^a v_l^b]_{a+b \leq k}$ have full rank and the masks Γ_j for the difference operators can be computed by (3.11).

The crucial difference between the functional and the parametric case is that for functional data, a global parameterization is always possible, i.e., a parameterization such that for all vertices $\mathbf{p}, \mathbf{q} \in \mathcal{U}_j \cap \mathcal{U}_l$ the condition $\|\mu_j(\mathbf{p}) - \mu_j(\mathbf{q})\| = \|\mu_l(\mathbf{p}) - \mu_l(\mathbf{q})\|$ is satisfied. We call this property *compatibility* of the local parameterizations μ_j . This definition allows neighboring parameterizations to differ by translations and rotations in the parameter plane. Both transformations have no influence on the value of $\|\text{diag}[\rho_0, \dots, \rho_{r-1}] \Gamma_j\|_2^2$ for discretizations of the rotational invariant operators in (3.2). If the local parameterizations μ_j are compatible then they can be patched together to form a global parameterization of the net.

In the general parametric case, however, the parameterizations $\mu_j(\mathcal{U}_j \cap \mathcal{U}_l)$ and $\mu_l(\mathcal{U}_j \cap \mathcal{U}_l)$ may not be as strictly related. At least, we require *asymptotic compatibility*. This means that if we have a specific scheme that computes the parameterizations μ_j for a refined net $\mathbf{S}_r N$ then for all $\mathbf{p}, \mathbf{q} \in \mathcal{U}_j \cap \mathcal{U}_l$

$$\frac{\|\mu_j(\mathbf{p}) - \mu_j(\mathbf{q})\|}{\|\mu_l(\mathbf{p}) - \mu_l(\mathbf{q})\|} \rightarrow 1 \quad (3.13)$$

as r increases to infinity. In other words, the more we refine the original mesh the more compatible are overlapping parameterizations. The motivation for this definition is the idea that, in order to let derivatives on a parametric surface have geometric relevance, a reparameterization in every point becomes necessary (cf. Sect. 4). However, since the limit surface is supposed to be smooth, these reparameterizations should locally become similar. In fact, the local metric of a surface can be estimated by projecting the neighborhood $\mathcal{U}(\mathbf{p})$ into the tangent plane at \mathbf{p} and the tangent plane is expected to vary continuously on a smooth surface.

To complete the discretization in the parametric case, it remains to define the weight coefficients w_j for the quadrature formula (3.6) which a priori does not make any sense because we have no global parameter domain Ω and therefore we do not have a proper integral to be approximated.

In the functional case there is a one to one correspondence between the triangles of the net N' and triangles in the parameter plane. Hence, integration over Ω , i.e., summation over parameter triangles, is equivalent to summation over the triangles of the mesh. In the parametric case, every triangle in \mathcal{T}' corresponds to several parameter triangles according the neighborhoods \mathcal{U}_j it belongs to. If we discretize the integral operator in the sense of summation over \mathcal{T}' (integration by area element), we basically have to estimate the contribution of each triangle $T \in \mathcal{T}'$ to the value of the energy functional.

The neighborhoods \mathcal{U}_j are usually associated with either the vertices, the edges, or the faces of the net N' . For example the neighborhood $\mathcal{U} := d_k(\mathbf{p})$ is centered around the vertex \mathbf{p} while $\mathcal{U} := d_k(\mathbf{p}) \cup d_k(\mathbf{q}) \cup d_k(\mathbf{r})$ with $T := [\mathbf{p}, \mathbf{q}, \mathbf{r}] \in \mathcal{T}'$ is centered around T .

Let $\Gamma_{\mathbf{p}} \in \mathbb{R}^{r \times m}$ be the difference operator associated with $\mathcal{U}_{\mathbf{p}}$ (and $\mu_{\mathbf{p}}$) for vertex-centered neighborhoods ($m = \#\mathcal{U}$), and $\Gamma_{\overline{\mathbf{p}, \mathbf{q}}}$ and Γ_T are defined for $\mathcal{U}_{\overline{\mathbf{p}, \mathbf{q}}}$ and \mathcal{U}_T analogously. The energy value for the sample corresponding to a neighborhood \mathcal{U} is computed by

$$\mathbf{E} := \text{trace}([\mathbf{p} \in \mathcal{U}]^T \Gamma^T \Gamma [\mathbf{p} \in \mathcal{U}]) \quad (3.14)$$

where $[\mathbf{p} \in \mathcal{U}]$ is a $(m \times 3)$ -matrix containing the coordinates of the vertices $\mathbf{p} \in \mathcal{U}$ in its rows.

Consider the triangle $T := [\mathbf{p}, \mathbf{q}, \mathbf{r}] \in \mathcal{T}'$. According to the three different situations $\mathcal{U}_{\mathbf{p}}$, $\mathcal{U}_{\overline{\mathbf{p}, \mathbf{q}}}$ and \mathcal{U}_T , its contribution to the total energy is given by

$$\frac{w}{3} (\mathbf{E}_{\mathbf{p}} + \mathbf{E}_{\mathbf{q}} + \mathbf{E}_{\mathbf{r}}) \quad \text{or} \quad \frac{w}{3} (\mathbf{E}_{\overline{\mathbf{p}, \mathbf{q}}} + \mathbf{E}_{\overline{\mathbf{q}, \mathbf{r}}} + \mathbf{E}_{\overline{\mathbf{r}, \mathbf{p}}}) \quad \text{or} \quad w \mathbf{E}_T$$

with the positive factor w reflecting the relative size of T compared to other triangles in \mathcal{T}' . Obviously, a common factor in all the weights does not affect the solution of the optimization. The size w can be chosen to be the average area of the triangles $\mu_j([\mathbf{p}, \mathbf{q}, \mathbf{r}])$ over the involved neighborhoods \mathcal{U}_j . Due to the asymptotic compatibility, these triangles do not differ much after sufficient refinement. Notice that we *cannot* simply set $w = \text{area}[\mathbf{p}, \mathbf{q}, \mathbf{r}]$ since the geometric location of the vertices is not known a priori: we *first* have to choose local parameterizations (including area element) and *then* solve the optimization problem to compute the positions of the new vertices. Therefore we have to estimate the area element *directly* from the local metrics μ_j .

The quadrature weights w_j in (3.6) are finally obtained by summing up the contributions to all triangles affected by a particular \mathcal{U}_j . For example $\mathbf{E}_{\mathbf{p}}$ for $\mathcal{U}_{\mathbf{p}} := d_k(\mathbf{p})$ affects all triangles $[\mathbf{q}_i, \mathbf{q}_{i+1}, \mathbf{p}]$ meeting at \mathbf{p} and therefore the w_j have to be proportional to $\sum_i \text{area}(\mu_{\mathbf{p}}([\mathbf{q}_i, \mathbf{q}_{i+1}, \mathbf{p}]))$.

Remark: The apparent advantage of difference methods to solve fairing problems (compared to Galerkin-type approaches) is that no globally consistent parameterization is necessary. All we need are local parameterizations for every neighborhood \mathcal{U}_j to derive the divided difference operators and a globally consistent estimate of the area element to weight the local samples correctly. As described above, once the local parameterizations μ_j are defined, the approximate area element can be derived under the assumption that the μ_j are locally close-to-isometric.

4 Local parameterizations for subdivided nets

We have to address the question how to choose *good* local parameterizations μ_j . Obviously, good parameterizations should approximate the isometric parameterization since in this case derivatives have a simple geometric interpretation. For example, the thin-plate functional

$$\text{TP}(\mathbf{f}) := \int \text{trace} \left((D\mathbf{f})^T (D\mathbf{f}) \right) \quad (4.1)$$

with (cf. (3.2))

$$D := \left[0, 0, 0, \frac{\partial^2}{\partial u^2}, \sqrt{2} \frac{\partial^2}{\partial u \partial v}, \frac{\partial^2}{\partial v^2} \right]^T$$

is equivalent to the total curvature

$$\text{TC}(\mathbf{f}) := \int \kappa_1^2 + \kappa_2^2 \quad (4.2)$$

if the first fundamental form of \mathbf{f} is the identity

$$\begin{pmatrix} \langle \mathbf{f}_u, \mathbf{f}_u \rangle & \langle \mathbf{f}_u, \mathbf{f}_v \rangle \\ \langle \mathbf{f}_v, \mathbf{f}_u \rangle & \langle \mathbf{f}_v, \mathbf{f}_v \rangle \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

However, a globally isometric parameterization does not exist in general and even a local reparameterization requires a priori knowledge about the resulting surface. Therefore we have to use heuristic methods to find good approximations of local isometric parameterizations μ_j .

There is a very subtle interplay between the choice of good local parameterizations and the resulting fair mesh: For the choice of the optimal μ_j we have to make assumptions about the resulting net which, on the other hand, itself depends on the μ_j . The choice of the μ_j should also draw into account that the notion of fair triangular nets has two aspects. First, *outer fairness* refers to small jumps of the normal vectors of adjacent faces. However, this does not provide a complete characterization. In addition, we have to guarantee *inner fairness* as well, i.e., the well-shaped-ness of individual triangular faces: There are good and bad triangulations for a set of points lying on the same fair surface (cf. Fig. 5).

4.1 Types of local topologies

We are not dealing with arbitrary triangular nets, but with nets which are generated by the application of a uniform refinement operator \mathbf{S}_r . This means that we have inner vertices, edge vertices and corner vertices forming a mesh with isolated singularities separated by regular regions. Since the size of the neighborhoods \mathcal{U}_j is defined in terms of topological distance (or number of vertices), every \mathcal{U}_j contains at most one singular (corner) vertex after sufficient refinement. We look for natural ways to define local parameterizations μ_j .

If a net N is a sufficiently refined approximation of a smooth parametric surface S then the vertices in a local neighborhood \mathcal{U}_j of a vertex \mathbf{p} approximately lie in a plane. In fact, for a local parameterization μ_j the triangles of vertices from \mathcal{U}_j essentially are affine images of the corresponding triangles in the parameter plane. This affine map is given by the first terms of the Taylor expansion of S at \mathbf{p} with respect to μ_j (especially if we minimize second order derivatives!). Although we do not know the Taylor expansion of the resulting surface S a priori, the parameterization for the vertices in \mathcal{U}_j should be chosen according to the local topology in order to promote *inner fairness*. Due to the special structure of the subdivision meshes, only three different constellations have to be distinguished.

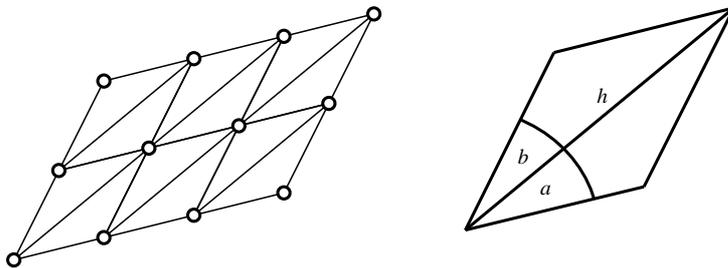


FIGURE 1. Parameterization of a uniform submesh (three degrees of freedom).

Consider a region \mathcal{U}_j which contains only inner vertices. Since there is nothing special about one particular vertex in the inner of \mathcal{U}_j , the parameterization should look the same around each of these vertices (i.e., in the 1-disc around each). A natural way to assign

parameter values in this case is therefore to use the knot-values of a (shift-invariant) affine grid in the parameter plane. This leaves three degrees of freedom (cf. Fig. 1). Notice that this choice for a parameterization is not arbitrary but it optimizes inner fairness in the sense that all parameter triangles are congruent.

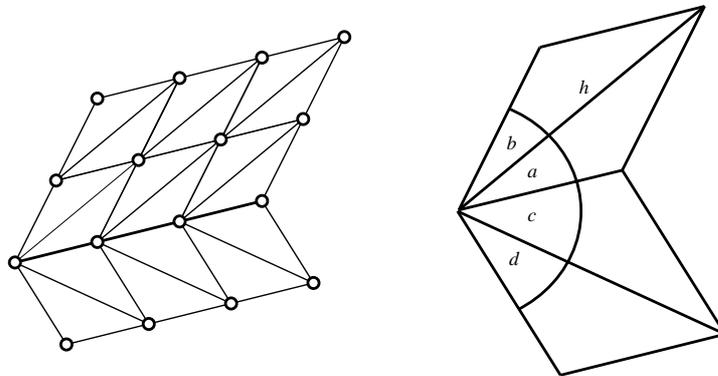


FIGURE 2. Parameterization of a semi-uniform submesh (five degrees of freedom).

If \mathcal{U}_j contains edge vertices then these vertices separate regions of inner vertices. Hence, a natural way to parametrize \mathcal{U}_j in this case is to allow a different affine grid in each of these regions. However, the step widths of these grids have to be compatible to avoid ambiguities (for edges connecting two edge vertices). The number of degrees of freedom is five in this case (cf. Fig. 2).

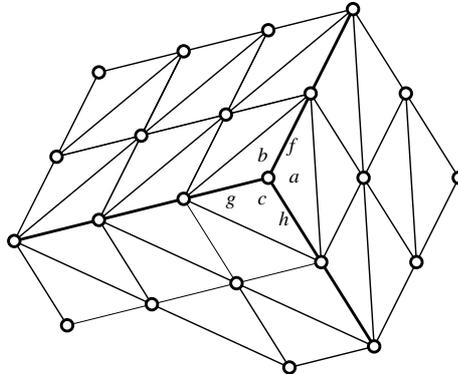


FIGURE 3. Parameterization of a semi-uniform submesh (five degrees of freedom in this case).

Finally, if \mathcal{U}_j contains a corner vertex \mathbf{p} then the adjacent edge vertices divide \mathcal{U}_j into a number of sectors according to the valence of \mathbf{p} . Again, in each of these sectors, an affine grid can be chosen where in addition to the constraints on the step widths, the angles around \mathbf{p} have to add up to 2π . It turns out that the number of degrees of freedom is two times the valence of \mathbf{p} minus one (cf. Fig. 3).

In each of the above constellations, we can pick one particular vertex of \mathcal{U}_j as a *crystal-seed* and extrapolate from it the whole parameterization μ_j . Defining the parameterization (the metric) at a particular vertex means assigning parameter values to its direct neighbors (1-disc). In case of a uniform grid this defines μ_j completely due to the shift invariance. Edge vertices can be seen as *cracks* in the crystal structure where two uniform regions

meet. Given the 1-disc around an edge vertex allows to propagate the crystal structure into both directions. In the last constellation (cf. Fig. 3), the parameter values for the vertices adjacent to the corner vertex are sufficient to reconstruct the whole mesh.

According to the number of degrees of freedom, different patterns for the crystal-seeds can be distinguished. Without loss of generality, we can assign the value $(0, 0)$ to the seed vertex \mathbf{p} and a value on the u -axis to one of its direct neighbors (since we restrict ourselves to rotationally invariant energy functionals).

If \mathbf{p} is an inner vertex then it has exactly 6 neighbors \mathbf{q}_l . The corresponding seed-pattern has three degrees of freedom α , β , and h

$$\mu_j([\mathbf{q}_l]) = \begin{pmatrix} \sin(\beta) h & 0 \\ \sin(\gamma) \cos(\alpha) h & \sin(\gamma) \sin(\alpha) h \\ \sin(\alpha) \cos(\gamma) h & \sin(\alpha) \sin(\gamma) h \\ -\sin(\beta) h & 0 \\ -\sin(\gamma) \cos(\alpha) h & -\sin(\gamma) \sin(\alpha) h \\ -\sin(\alpha) \cos(\gamma) h & -\sin(\alpha) \sin(\gamma) h \end{pmatrix} \quad (4.3)$$

where $\gamma = \alpha + \beta$. The area of the 1-disc $\mu_j([\mathbf{q}_l])$ in the parameter plane is $\text{area}(\mu_j([\mathbf{q}_l])) = 3 h^2 \sin(\alpha) \sin(\beta) \sin(\gamma)$.

If \mathbf{p} is an edge vertex (also valence 6), two additional parameter $\bar{\alpha}$ and $\bar{\beta}$ are introduced

$$\mu_j([\mathbf{q}_l]) = \begin{pmatrix} \sin(\beta) h & 0 \\ \sin(\gamma) \cos(\alpha) h & \sin(\gamma) \sin(\alpha) h \\ \sin(\alpha) \cos(\gamma) h & \sin(\alpha) \sin(\gamma) h \\ -\sin(\bar{\beta}) \bar{h} & 0 \\ \sin(\bar{\alpha}) \cos(\bar{\gamma}) \bar{h} & -\sin(\bar{\alpha}) \sin(\bar{\gamma}) \bar{h} \\ \sin(\bar{\gamma}) \cos(\bar{\alpha}) \bar{h} & \sin(\bar{\gamma}) \sin(\bar{\alpha}) \bar{h} \end{pmatrix} \quad (4.4)$$

with $\bar{\gamma} = \bar{\alpha} + \bar{\beta}$ and $\bar{h} = h \sin(\beta) / \sin(\bar{\beta})$. Here, the area is

$$\text{area}(\mu_j([\mathbf{q}_l])) = \frac{3}{2} h^2 [\sin(\alpha) \sin(\beta) \sin(\gamma) + \sin(\bar{\alpha}) \sin(\bar{\beta}) \sin(\bar{\gamma})].$$

In the most general case, \mathbf{p} is a corner vertex with valence r and the free parameters are $\alpha_1, \dots, \alpha_r$ and h_1, \dots, h_r :

$$\mu_j([\mathbf{q}_l]) = \begin{pmatrix} \vdots & \vdots \\ h_l \cos\left(\sum_{i=1}^{l-1} \alpha_i\right) & h_l \sin\left(\sum_{i=1}^{l-1} \alpha_i\right) \\ \vdots & \vdots \end{pmatrix} \quad (4.5)$$

with the only constraint that $\sum_l \alpha_l = 2\pi$. The area in the parameter plane is $\text{area}(\mu_j([\mathbf{q}_l])) = \frac{1}{2} \sum_i h_i h_{i+1} \sin(\alpha_i)$.

4.2 Approximating a local isometric parameterization

The simplest way to describe the local reparameterization of a smooth surface S such that isometry is obtained at a certain point \mathbf{p} , is to assign parameter values in a neighborhood U_j of \mathbf{p} according to the coordinates of the orthogonal projection into the tangent-plane $H_{\mathbf{p}}$ of S at \mathbf{p} with respect to an orthonormal frame spanning $H_{\mathbf{p}}$. Hence, to approximate

the values of an isometric parameterization at the neighboring vertices $\mathbf{p}_i \in \mathcal{U}_j \subset \mathcal{P}'$ it is sufficient to know how the projection of \mathcal{U}_j into the tangent plane $H_{\mathbf{p}}$ looks like. Therefore, any heuristic method to estimate the tangent plane at a point \mathbf{p} of a triangular net yields an estimation of a local isometric parameterization.

The local energy value (3.14) does not change, if we apply an orthonormal transformation $Q \in \mathbb{R}^{3 \times 3}$ to the row vectors $\mathbf{p} \in \mathcal{U}$ (rotation), i.e.,

$$\mathbf{E} = \text{trace} \left((\Gamma[\mathbf{p} \in \mathcal{U}]) Q Q^T (\Gamma[\mathbf{p} \in \mathcal{U}])^T \right) = \text{trace} \left((\Gamma[\mathbf{p} \in \mathcal{U}] Q) (\Gamma[\mathbf{p} \in \mathcal{U}] Q)^T \right).$$

Hence, without loss of generality, we may assume that the tangent plane at \mathbf{p} is the xy -plane of a local coordinate system and this takes us back to the functional case: The three components of the vector valued function \mathbf{f} can be optimized separately (cf. (3.12)) and the projection of the (locally) optimal solution into the tangent plane $H_{\mathbf{p}}$ produces the identity.

This can be exploited to enforce inner fairness: A good choice for the parameterizations μ_j is to fix the values for neighboring vertices exactly as their projection into the tangent plane *should* look like. Fig. 4 shows how this allows to influence the solution of the optimization in a predictable way: For every neighborhood \mathcal{U}_j the minimum energy is achieved if two coordinate functions are the identity (and the third being from the energy norm's kernel). Hence, the global minimum which balances the local minima, approximates the identity locally, i.e., the optimal net looks locally 'as similar as possible' to its parameterization. To some extent we do not have to approximate the metric of the resulting surface but conversely, due to the energy minimization, the metric of the resulting surface will approximate what we prescribed with the μ_j .

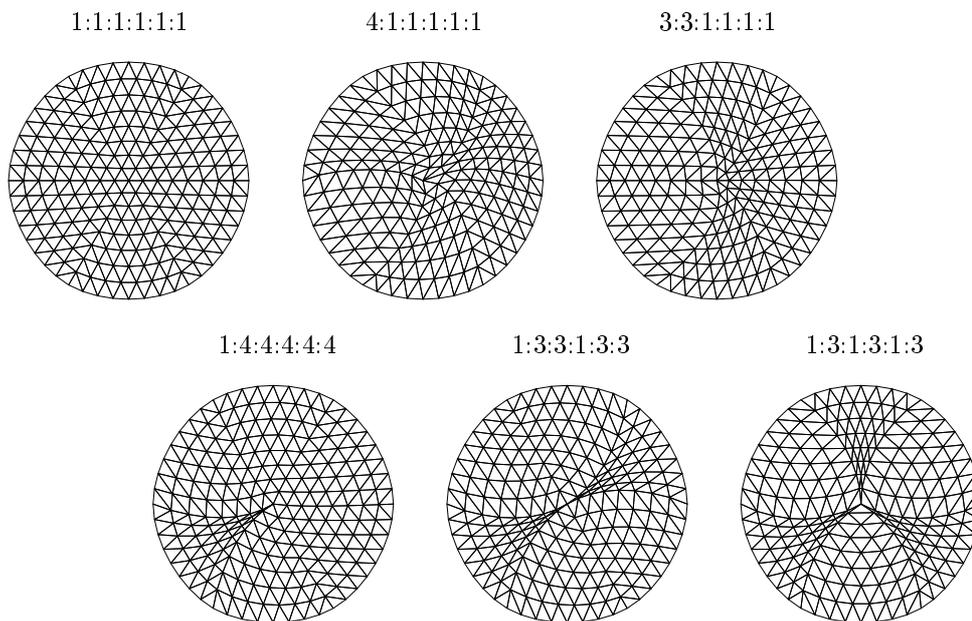


FIGURE 4. Result of minimizing the thin-plate functional for different local parameterizations at the origin (center vertex). The angles of the solution in the 1-disc around the origin are influenced by the prescribed ratios in the parameterization.

Let us recollect the important features for the local parameterizations μ_j : Since after refinement and smoothing, the triangles of the mesh locally look similar to the corresponding triangles in the parameter plane, the local parameterizations have to be *fair*. To

achieve this, the μ_j should be defined according to the three different constellations of local topologies (cf. Sect. 4.1). Further, since on a smooth surface, the tangent plane varies continuously, the local parameterizations have to be *asymptotically compatible*. This is easy to accomplish since the sizes of the local neighborhoods are measured in topological distances and therefore the ‘difference’ between the parameterizations at two vertices \mathbf{p} and \mathbf{q} being directly connected in the original net N can be equally distributed among more and more intermediate parameterizations along the edge $\overline{\mathbf{p}\mathbf{q}}$ as the refinement level r increases.

4.3 Constructing the parameterizations μ_j

Now we discuss simple schemes to construct local parameterizations which satisfy all the requirements of the last sections, namely that the μ_j should be locally isometric and with increasing resolution the local parameterizations of overlapping neighborhoods should become asymptotically compatible (cf. (3.13)). The problem to be solved is, given an original triangular net N , find local parameterizations μ_j for the neighborhoods \mathcal{U}_j in the refined net $\tilde{\mathbf{S}}_r N$ without knowing the exact position of the newly inserted vertices.

A very intuitive method to define those parameterizations takes another refined net $\tilde{N} := \tilde{\mathbf{S}}_r N$ obtained by applying an appropriate (stationary) subdivision operator $\tilde{\mathbf{S}}_r$ (e.g. the butterfly-scheme [DGL90]). The net \tilde{N} serves as a first approximation to the fair net. Its tangent planes can be estimated based on weighted averages of the adjacent edges or normals. The orthogonal projections of the topological neighborhoods $\mathcal{U}_j \in \tilde{\mathcal{P}}$ into the respective tangent planes yield parameterizations μ_j which, according to Sect. 4.2, are discretizations of local isometric parameterizations.

The quality of the resulting parameterization depends on the fairness of the net \tilde{N} generated by the underlying subdivision scheme $\tilde{\mathbf{S}}_r$. Notice that the vertex locations of \tilde{N} can also be used as *starting values* for the iterative solution of (3.8). This approach is similar to a discrete version of the data dependent functionals of [GLW96] where a reference surface is used to estimate the metric of the solution.

However, applying a stationary refinement scheme makes it impossible to guarantee the regularity, e.g., the absence of self intersections, in the refined mesh. This is a simple consequence of the affine invariance of the refinement operator. Hence, a scheme that bases the construction of the local parameterizations on computations in the net $\tilde{\mathbf{S}}_r N$ is not safe since any lower bound α_ε for angles of triangles in the parameter plane may be violated. This affects numerical stability in the construction of the difference operators Γ_j .

Further, this general construction makes it very difficult to analyze the matrix of the resulting linear system (3.8). We therefore restrict the μ_j to the three types described in Sect. 4.1. For these, it is sufficient to have a scheme generating a crystal seed for every vertex in \mathcal{P}' (cf. (4.3) - (4.5)). The special type of these seeds is determined by the type of the vertices. In Sect. 5 we will show how this restriction allows to prove the existence of a unique solution of the optimization problem.

One construction which satisfies these requirements uses a subdivision scheme $\tilde{\mathbf{S}}_r$ to estimate the tangent planes. But then the μ_j are obtained by projecting the vertices of $\mathbf{L}_r N$ into those planes (cf. (2.1)). The ‘piecewise linearity’ of \mathbf{L}_r guarantees that the templates of the resulting crystal seeds automatically correspond to the type of the vertices.

In Sect. 4.2 we concluded that the local parameterizations should be chosen such that they look like a desirable orthogonal projection of a fair net into its pseudo tangent plane. If we define the parameterizations at the vertices of N by projecting the vertices of $\mathbf{L}_r N$, the results – although smooth – are not necessarily fair with respect to inner fairness (cf. Fig. 5).

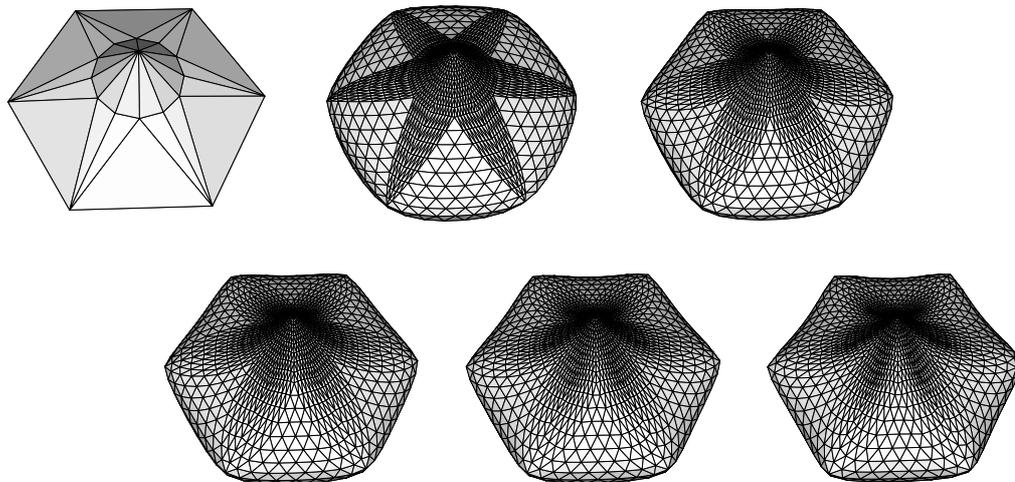


FIGURE 5. Interpolation of a simple object (upper left) with differently equalized local parameterizations at the corner vertices. In the first refined net, the parameterization of the corner vertices is obtained by projection into an estimated tangent plane. The last net results from using symmetric crystal seeds. The three other nets are obtained by taking linear combinations of the extremal seeds.

In some practical experiments with the discrete fairing, it turned out that a modification of the local parameterizations μ_j by equalizing edge-lengths and angles around each vertex may occasionally lead to better results. Generally, there seems to be a trade-off between inner and outer fairness (cf. Fig. 5).

Yet another way to define the tangent planes is to estimate them at the vertices of the original unrefined net N only and derive them for the new vertices of N' by interpolation of the normal vectors across triangles of N , e.g., by cubic Hermite interpolation. This is motivated by the notion of a continuously varying tangent plane on a smooth surface.

We can also interpolate between vertices of the original net without referring to geometric information, i.e., we can *interpolate the parameterizations μ_j directly*. We start by estimating tangent planes and local parameterizations at the vertices in the original net N . These vertices are corner vertices in the refined net N' . A seed corresponding to a corner vertex uniquely defines adjacent edge- and inner-type seeds.

Let \overline{pq} be an edge of the original net N . The seeds of μ_p and μ_q both define an edge-type seed associated with \overline{pq} . Since these edge-type seeds are topologically isomorphic (and orientation is well-defined) the edge-type seeds for intermediate vertices subdividing \overline{pq} can be constructed by simple interpolation (cf. Fig. 6).

The same construction can be carried out across every triangular face $T = [p, q, r] \in \mathcal{T}$. The seeds of μ_p , μ_q and μ_r determine inner-type seeds for T which can be interpolated according to the barycentric indices of the inserted vertices $\mathbf{q}_{u,v,w}$ (with $u, v, w \geq 1$). The orientation can be derived by finding corresponding directions in the regular submesh which subdivides $[p, q, r]$. Notice that we are interpolating the parameter seeds *directly* and do not perform interpolation in \mathbb{R}^3 .

The motivation for this construction, again, is that on a smooth surface the isometric reparameterization varies continuously. By this interpolation scheme, we somewhat distribute the difference between the parameterizations (metrics) at p and at q over all local neighborhoods introduced during refinement. Obviously, the interpolation of local

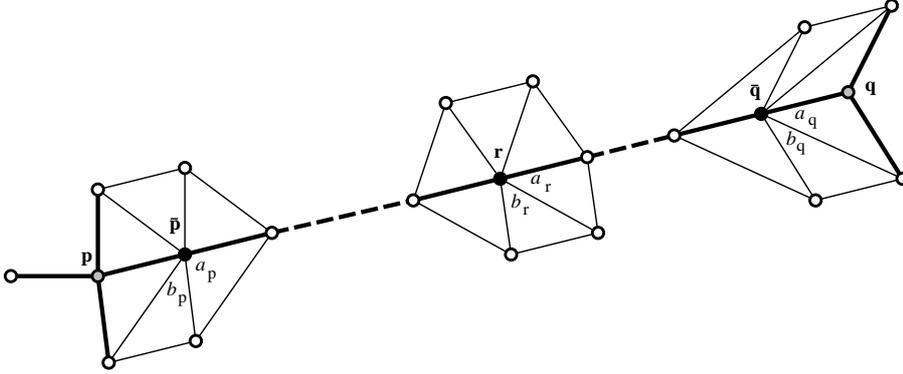


FIGURE 6. Definition of local parameterizations by interpolation. The seeds at the corner vertices \mathbf{p} and \mathbf{q} determine the edge-type seeds at $\bar{\mathbf{p}}$ and $\bar{\mathbf{q}}$ respectively. The seed at \mathbf{r} is then given, e.g., by $\alpha_{\mathbf{r}} := u \alpha_{\mathbf{p}} + (1 - u) \alpha_{\mathbf{q}}$ and $\beta_{\mathbf{r}} := u \beta_{\mathbf{p}} + (1 - u) \beta_{\mathbf{q}}$ for some appropriate parameter $u \in [0, 1]$.

parameterizations ensures asymptotic compatibility.

In the above description, the term ‘interpolation’ is used generically. The local parameterizations can be computed, e.g., by interpolating the vector of degrees of freedom (angles and stepwidth) or — not equivalently — by interpolating the actual parameter values (vector of coordinates). Also, we restricted our description to interpolation schemes that use direct neighbors only. Higher order schemes could be applied as well.

5 Thin-plate-splines

We now give a detailed instance of the general framework described in the previous sections: the discretization of the minimization of total curvature (4.2) which can be approximated by a combination of squared second order derivatives (4.1) if the underlying parameterization is close to isometric. Since we do not exploit the special structure of the matrix $\text{diag}[\rho_i]$ in the definition of the differential operator D , every statement in this section holds for *any* discrete fairing scheme that measures fairness based on squares of second order differences (derivatives).

At first we have to choose the neighborhoods \mathcal{U}_j for the construction of divided difference operators. Since we have to approximate second order derivatives only, the 1-discs $d_1(\mathbf{p})$ around every vertex $\mathbf{p} \in \mathcal{P}$ are appropriate. In regular regions of the net, these contain seven vertices and a least norm solution can be found by (3.11). A unique solution exists if the minimum angle in the parameter triangles is bounded below by a positive constant α_ε . Problems occur at singular vertices of valence three or four where the difference operator masks are over-determined. In these special cases we have to include more vertices into the neighborhoods (cf. Fig 7).

Following the outline of Sect. 4.3, we have to estimate good parameterizations for the 1-discs around the original (corner-)vertices first. All other local parameterizations in the refined net N' are then obtained by interpolation.

Having constructed the difference operator masks Γ_j (3.11), the next step is to determine the quadrature coefficients w_j . According to Sect. 3.4, we simply set them to the area that is covered by $\mu_j(\mathcal{U}_j)$. Notice that the asymptotic compatibility of the μ_j and the restriction to the three types of local parameterizations (4.3) - (4.5) imply that for overlapping neighborhoods \mathcal{U}_j and \mathcal{U}_i the *ratio* of $\text{area}(\mu_j(\mathcal{U}_j \cap \mathcal{U}_i))$ and $\text{area}(\mu_i(\mathcal{U}_j \cap \mathcal{U}_i))$

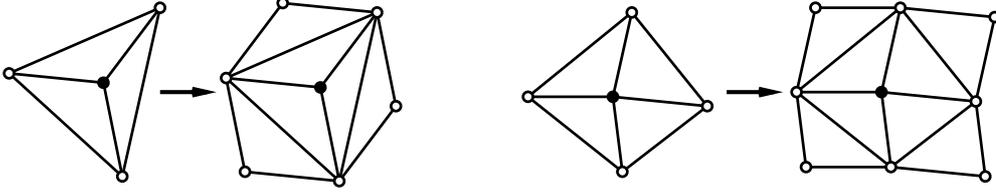


FIGURE 7. Expanding the neighborhoods \mathcal{U}_j in the vicinity of low-valence singular vertices.

converge. This makes the optimization problem well-defined since a common factor to all weights w_j does not affect the solution.

It remains to verify condition (3.9) or equivalently $\ker(\tilde{\Gamma}) = \{0\}$. This is not trivial and we have to exploit the special structure of the local parameterizations μ_j . The particular interpolation scheme by which the local parameterizations at the new vertices $\mathbf{q}_{u,v,w}$ are computed, turns out to be not important for this analysis as long as the seeds are of the forms (4.3) - (4.5).

We have to show that, except for the trivial net Z which completely collapses at the origin, there exists no net \tilde{N} topologically isomorphic to N' which has zero energy and all corner vertices lying at the origin. Such a net could be added to a minimal solution without changing its energy nor its interpolation properties and thus making the optimization problem ill-posed.

Since the optimization in the parametric case can be performed separately for each coordinate function, we can restrict the following considerations to scalar valued meshes. Consider a single 1-disc \mathcal{U}_j around an inner vertex. According to the parameterization (4.3), the least norm masks Γ_j of (3.11) for the second order difference operators can be given explicitly by

$$M_{xx} := \frac{1}{w^2} \begin{pmatrix} 0 & 0 & 0 \\ v^2 & -2v^2 & v^2 \\ 0 & 0 & 0 \end{pmatrix} \quad (5.1)$$

$$M_{xy} := \frac{1}{w^2} \begin{pmatrix} -q & q & -p \\ -p & 2p & -p \\ q & -q & -q \end{pmatrix} \quad (5.2)$$

$$M_{yy} := \frac{1}{w^2} \begin{pmatrix} su & -tu & st \\ st & -2(st+u^2) & st \\ -tu & su & su \end{pmatrix} \quad (5.3)$$

where $\gamma = \alpha + \beta$, $s = \cos(\alpha) \sin(\gamma)$, $t = \sin(\alpha) \cos(\gamma)$, $u = \sin(\beta)$, $v = \sin(\alpha) \sin(\gamma)$, $w = \sin(\alpha) \sin(\beta) \sin(\gamma)$, $p = \frac{1}{2}(s+t)v$, and $q = \frac{1}{2}uv$. Due to the symmetry of the weight coefficients, the kernel of the local operator $\Gamma_j = [M_{xx}, M_{xy}, M_{yy}]$ can be represented by the simple basis

$$\text{span} \left\{ \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ -1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & -1 \end{pmatrix} \right\}.$$

Notice that this basis is valid for every parameterization of the form (4.3) and does not depend on the actual angles and lengths in $\mu_j(\mathcal{U}_j)$. Obviously, the kernel of Γ_j has *four*

dimensions while the kernel of the corresponding continuous differential operator has only three dimensions (the set of affine functions). This bigger kernel makes the analysis of the unique solvability (cf. (3.9)) necessary.

Without loss of generality we assume that the net N' is generated by a subdivision operator \mathbf{S}_r with $r \geq 4$. For smaller r more special cases have to be considered. Let \mathcal{U}_j , \mathcal{U}_k , and \mathcal{U}_l be three overlapping 1-discs corresponding to adjacent inner vertices. From the over-determination of vertex values (according to the individual kernels of Γ_j , Γ_k , and Γ_l respectively), it follows that the kernel of the combined operator $[\Gamma_j, \Gamma_k, \Gamma_l]$ is only three dimensional (cf. Fig. 8).

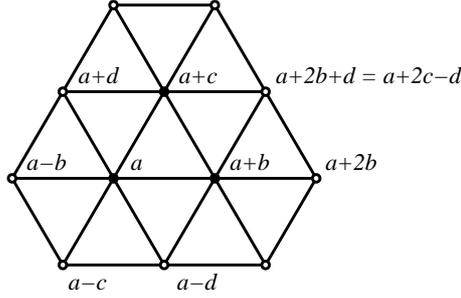


FIGURE 8. Constructing a basis for the kernel of $\tilde{\Gamma}$.
The remaining degrees of freedom are a , b , and d .

The condition $a + 2b + d = a + 2c - d$ implies $c = a + d$ and by further propagating this condition over other neighboring 1-discs of inner-vertex type, it turns out that the kernel of the combined operator $[\Gamma_{u,v,w}]_{u,v,w \geq 1}$ covering the 1-discs $\mathcal{U}_{u,v,w}$ around all inner vertices belonging to the same initial triangular face, consists of uniform samples of an affine function. The same holds for any other submesh of inner vertices.

The local kernels (one for each original face) are linked together via the 1-discs around edge vertices. Since the kernel conditions within each inner region imply that the vertex values are samples of an affine function there is only one degree of freedom, x , left in the 1-disc around an edge vertex \mathbf{e} . The others are determined since all edge vertices lie on the intersection of two neighboring affine functions. The value x has to be chosen such that we stay also within the kernel of $\Gamma_{\mathbf{e}}$. Fig. 9 depicts the situation.

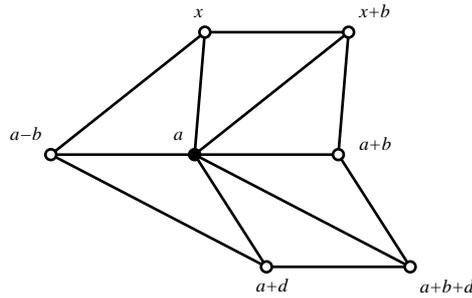


FIGURE 9. Constraints for the kernel vector in the vicinity of an edge vertex \mathbf{e} (solid dot).

Obviously, the only solution for x is the one that makes the vertex values samples of an affine function with respect to the edge vertex's local parameterization. Any other x would imply a quadratic term in the interpolating polynomial and thus cannot lie in the kernel of $\Gamma_{\mathbf{e}}$ (cf. the construction of the difference operators by solving a Vandermonde

system in Sect. 3.2). This is true since three parallel lines can always be interpolated by a quadratic bivariate polynomial.

The solution x fixes *all* degrees of freedom in the link between two regions of inner vertices since each such region is uniquely determined by one of its parameter triangles (crystal seeds). If the local parameterizations for two edge vertices along the same edge of the original mesh N differ by more than an affine map then no global solution exists along that edge and the kernel of the global functional Γ is empty.

However, if the edge-vertex parameterizations are related by an affine map² then the differences between vertex values of the kernel vector along edges that connect two edge vertices is constant (uniform samples of an affine function; the value b in Fig. 9). At both ends of such a chain of ‘edge-edges’ there are corner vertices whose vertex value has to be zero in order not to violate the interpolation condition. Considering these restrictions along the three boundaries of a regular submesh corresponding to a single triangular face of the original net, it turns out that every local kernel vector has to consist of uniform samples of an affine function with three roots not lying on a straight line. The only such function is identically zero and hence the kernel of the matrix $\tilde{\Gamma}$ is trivial. \square

5.1 Higher order smoothness

The thin-plate variational scheme discussed in the last section produces interpolating nets of high quality (cf. the example in Sect. 7). However from a geometric point of view, it could make more sense to minimize the *variation* of curvature, i.e., to distribute the curvature equally over the surface instead of minimizing it [MS92]. The changes of curvature are measured by third derivatives.

Therefore the minimization of a discrete version of the energy functional (3.1) based on (3.2) with $k = 3$ can be expected to lead to even more pleasant shapes. However, both the construction of the difference operators as well as the solution of the resulting linear system are computationally more involved. This is the well-known trade-off between surface quality and computational complexity.

6 Efficient computation

The computation of the vertex positions in a fair net requires the solution of the large sparse system (3.8). Such systems are most efficiently solved by iterative algorithms [Hac91]. Due to the geometric background of the problem, it is possible to find very good starting values for the iteration. Every step of the iterative scheme can be considered as a smoothing step taking the vertices closer to the fair solution of the optimization problem.

Typically for such low-pass filter type iterations, the matrices of these systems are asymptotically ill conditioned, i.e., the spectral radius ρ of the corresponding iteration matrix in a Gauß-Seidel- or Jacobi-scheme converges to $\rho = 1$ like $O(1 - r^{-2})$ where r is the order of the refinement operator \mathbf{S}_r . The reason for such behavior is that in a single iteration step only local information (one row of the sparse matrix) is used to adjust the position of a vertex. Hence, local (high frequency) oscillations are smoothed out rather quickly while low frequency errors decrease slowly.

Fig. 11 shows the oscillations of eigenvectors corresponding to a Gauß-Seidel iteration matrix for the iterative thin-plate energy minimization. The initial interpolation data is a regular tetrahedron T and the values are color coded on the surface of the minimum energy solution. The shading represents the absolute value of the Laplacian (high-pass) of the actual kernel vector values in order to emphasize the oscillatory behavior.

²Such a situation occurs, e.g., when we consider the graph of a function as a special parametric surface.

The pictures in Fig. 11 are based on the 18th order refinement $N' := \mathbf{S}_{18}T$ of the tetrahedron. The refined net N' contains 650 vertices, four of which are fixed due to the interpolation conditions. For the application of the Gauß-Seidel iteration, the matrix $A := \tilde{\Gamma}^T \tilde{\Gamma}$ has to be split into its lower triangular part L (including the diagonal) and the upper triangular part R . The contraction rate of the error is then bounded by

$$\rho(I - L^{-1}A) = 1 - \lambda_{\min}(L^{-1}A).$$

We computed the eigenvectors of $L^{-1}A$ and sorted them according to their corresponding eigenvalues in increasing order. The first vectors in this list span the subspaces where the convergence of the iterative scheme is slowest. In Fig. 11 we show the vectors 1, 20, 50, 100, 200, 300, 450, and 646 of this list. Apparently, frequency and amplitude of the oscillation increase.

A practical consequence of this special eigenstructure of the iteration matrix is that (by using a naive solving algorithm) locally smooth meshes can be found quite efficiently while convergence to the exact fair solution of the underlying variational problem cannot be achieved in a reasonable time. However, low-pass filter characteristics of the iteration matrix in a solving scheme is the classical situation where multi-grid schemes are expected to accelerate the iterative solver significantly [Hac85]. The justification for this statement is that, due to its special eigenstructure, the iterations on lower refinement levels reduce the errors in those subspaces which are doomed to slow convergence on the finer level.

The multi-resolutional decomposition of (3.8) naturally arises from the subdivision topology of the refined net N' . If the optimization is performed on $N' = \mathbf{S}_r N$ then a maximum sequence of embedded levels of resolution is obtained from factoring r into its prime factors r_i . The most simple choice is $r_i := 2$ and $r = 2^m$.

A multi-grid algorithm consists of two ingredients: a basic iterative solver with smoothing characteristics and prolongation/restriction operators P_i/R_i to switch between levels. Since we can exploit the geometric coherence in the mesh data, it is easy to find good (smooth) starting values, e.g., we can use appropriate stationary subdivision operators $\tilde{\mathbf{S}}_{r_i}$ to obtain the initial net $\tilde{N} = \tilde{\mathbf{S}}_{r_n} \dots \tilde{\mathbf{S}}_{r_1} N$.

The action of a simple V-cycle multi-grid scheme Ψ on the i -th refinement level can be described by $\Psi_i = \Phi P_i \Psi_{i-1} R_i \Phi$ where Φ is the basic iterative solver (smoother). In our special situation, pre-smoothing is not necessary because the initial vertex positions in \tilde{N} usually are sufficiently smooth. A natural choice for the restriction operator is $R_i := \tilde{\mathbf{S}}_{r_i}^{-1}$ and for the prolongation $P_i := \tilde{\mathbf{S}}_{r_i}$. Classically, R_i and P_i have to be *adjoint* operators with respect to the underlying error norm. However, our special choice makes the discrete fairing algorithm much simpler and does not affect the convergence behaviour significantly.

With these assumptions, unfolding the whole recursive V-cycle algorithm leads to

$$N' := \Psi_n N = \left(\prod_{i=1}^n \Phi \tilde{\mathbf{S}}_{r_i} \right) N,$$

i.e., stationary subdivision $\tilde{\mathbf{S}}_{r_i}$ alternates with iterative smoothing Φ (e.g. Gauß-Seidel). Hence, an appropriate adaption of the multi-grid paradigm to the iterative fairing of triangular nets reduces to the back-tracking phase (back leg) of a V-cycle multi-grid scheme.

7 Conclusions

To conclude this paper we show an example surface to give an impression of the quality of the resulting interpolants. In Fig. 10, we show the original mesh (left) and the refined

surface (right). We used the subdivision operator $\mathbf{S}_8 = \mathbf{S}_2^3$ under minimization of the thin-plate functional. The refinement is sufficient to produce a visually smooth surface. The multi-grid algorithm to solve the corresponding sparse linear system took about 90 sec. on a SGI Onyx to compute the refined mesh with about 43500 vertices.

The proposed discrete fairing scheme can easily be generalized to *open* triangular meshes where special treatment of the boundaries becomes necessary. Following [Kob96b] this also enables the scheme to produce C^0 features like creases or cusps within an otherwise smooth surface.

For some *real world* applications, the interpolation of the vertices of the given net might be a too strict requirement since the measured vertex positions are subject to noisy errors. Hence, future work should investigate a modification of the fairing scheme by relaxing interpolation constraints to approximation constraints satisfying prescribed error tolerances.

This work was supported by a scholarship granted by the science section of NATO and organized by the DAAD. I am very grateful to Prof. Carl de Boor who made this project possible and I would like to thank him for the very helpful discussions on variational methods.

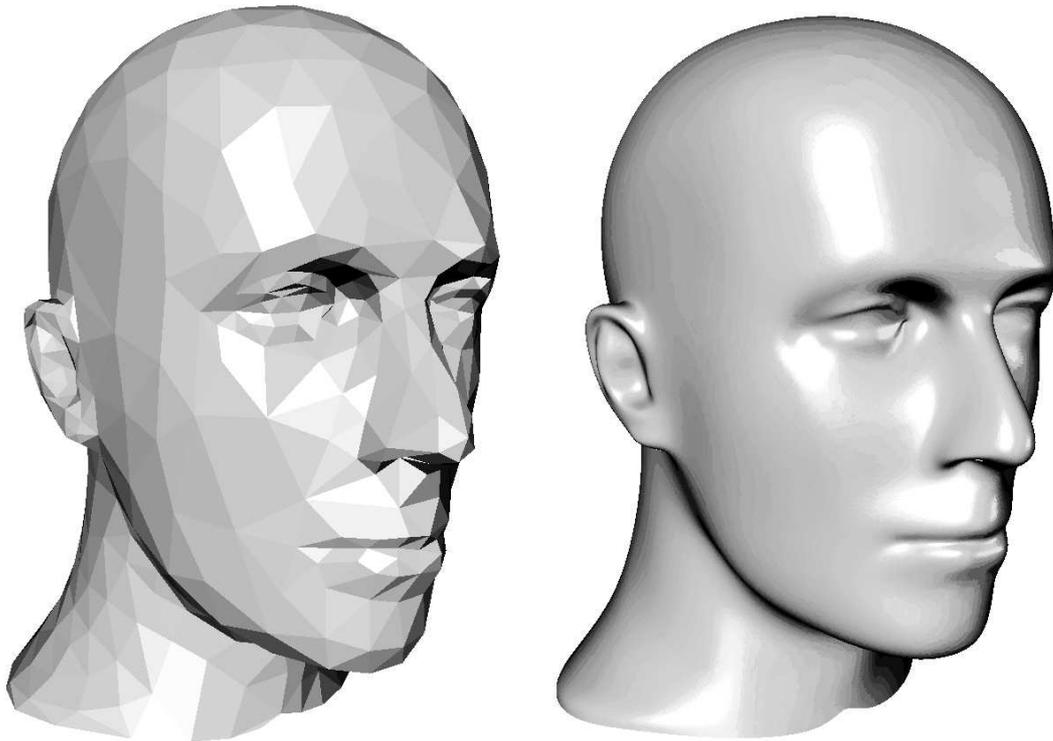


FIGURE 10. Original triangular mesh N and the refined net $\mathbf{S}_8 N$ obtained by minimization of a discrete thin-plate energy. Both pictures are rendered with flat shading.

References

- [CG91] G. Celniker / D. Gossard, *Deformable curve and surface finite elements for free-form shape design*, ACM Computer Graphics, 25:257–265, 1991
- [DGL90] N. Dyn / J. Gregory / D. Levin, *A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control*, ACM Trans. Graph. 9 (1990), pp. 160–169
- [Duc79] J. Duchon, *Splines minimizing rotation invariant semi-norms in Sobolev spaces*, W. Schempp, K. Zeller, eds., Multivariate Approximation Theory, Birkhäuser 1979, pp. 85–100
- [Els70] L. Elsgolc, *Variationsrechnung*, Teubner Verlag Stuttgart 1970
- [Gre94] G. Greiner, *Variational design and Fairing of Spline Surfaces*, Computer Graphics Forum, 13(3):143–154, 1994
- [GLW96] G. Greiner / J. Loos / W. Wesselink, *Data Dependent Thin Plate Energy and its Use in Interactive Surface Modeling*, Computer Graphics Forum, 1996
- [Hac85] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer Verlag, Berlin, 1985
- [Hac91] W. Hackbusch, *Iterative Lösung großer schwach besetzter Gleichungssysteme*, Teubner Verlag 1991, Stuttgart
- [HKD93] M. Halstead / M. Kass / T. DeRose, *Efficient Fair Interpolation Using Catmull-Clark Surfaces*, ACM Computer Graphics, 27:35–44, 1993
- [Kob96a] L. Kobbelt, *A Variational Approach to Subdivision*, CAGD 13 (1996), pp. 743–761
- [Kob96b] L. Kobbelt, *Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology*, Computer Graphics Forum 15 (1996), Eurographics '96 Conference Issue, pp. 409–420
- [MS92] H. Moreton / C. Séquin, *Functional Optimization for Fair Surface Design*, ACM Computer Graphics, 1992, pp. 167–176
- [Pow94] M. Powell, *The Uniform Convergence of Thin Plate Interpolation in Two Dimensions*, Numer. Math. 68 (1994), pp. 107–128
- [Sap94] N. Sapidis ed., *Designing Fair Curves and Surfaces*, SIAM, Philadelphia, 1994
- [ZSS96] D. Zorin, P. Schröder, W. Sweldens, *Interpolating Subdivision for Meshes with Arbitrary Topology*, ACM Computer Graphics, 1996, pp. 189–192

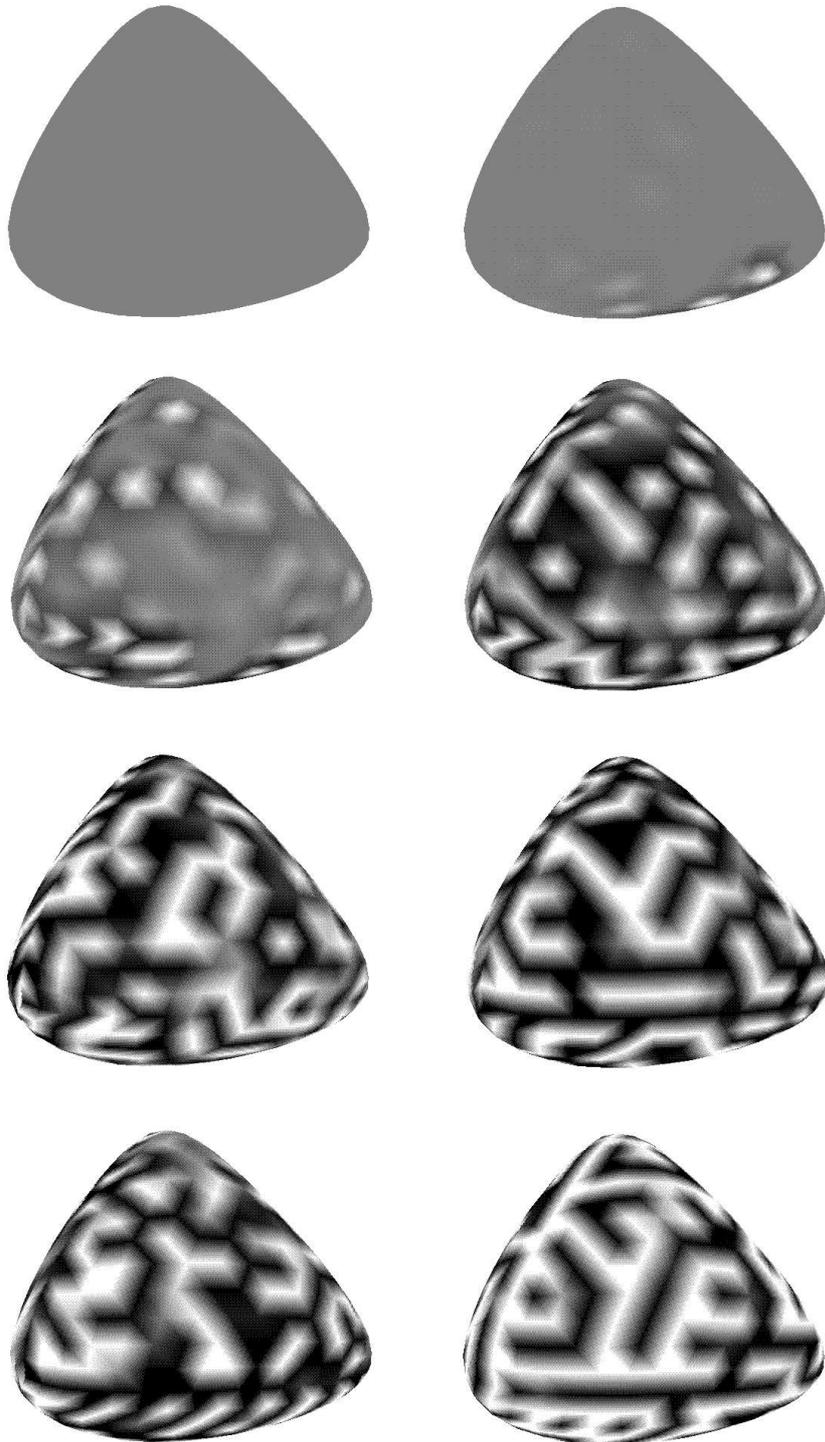


FIGURE 11. Color coded Laplacian of the eigenvectors of the Gauß-Seidel iteration matrix corresponding to the minimization of the thin-plate energy.