Weierstraß-Institut
für Angewandte Analysis und Stochastik
Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

A novel surface remeshing scheme via higher dimensional embedding and radial basis functions

Franco Dassi, Patricio Farrell, Hang Si

submitted: May 24, 2016

Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: franco.dassi@wias-berlin.de
patricio.farrell@wias-berlin.de
hang.si@wias-berlin.de

No. 2265
Berlin 2016

2010 *Mathematics Subject Classification.* 65M50.

Key words and phrases. Anisotropic meshes, radial basis function RBF, mesh optimization, implicit surfaces, geometry processing.

The work of Franco Dassi was partially supported under the Leibniz - DAAD Research Fellowship 2014.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Abstract

Many applications heavily rely on piecewise triangular meshes to describe complex surface geometries. High-quality meshes significantly improve numerical simulations. In practice, however, one often has to deal with several challenges. Some regions in the initial mesh may be overrefined, others too coarse. Additionally, the triangles may be too thin or not properly oriented. We present a novel mesh adaptation procedure which greatly improves the problematic input mesh and overcomes all of these drawbacks. By coupling surface reconstruction via radial basis functions with the higher dimensional embedding surface remeshing technique, we can automatically generate anisotropic meshes. Moreover, we are not only able to fill or coarsen certain mesh regions but also align the triangles according to the curvature of the reconstructed surface. This yields an acceptable trade-off between computational complexity and accuracy.

1 Introduction

Countless numerical methods need to transfer information from a continuous domain to discrete points on a grid. Most commonly this problem arises when solving partial differential equations numerically but also it appears in the context of surface approximation [16, 14, 24, 8] or medical image reconstruction [11].

Hence, finding *optimal* grids is of uttermost importance. Obviously, optimal can mean many things. However, two desirable features stand out. On the one hand, our grid should be built in such way that the data we are interested in (for instance the solution of a partial differential equation) is approximated fairly accurate. That is, we want to grasp finer details (for example along boundary layers) as well as large-scale variations. On the other hand, we want to be able to efficiently compute the discrete approximation and only use nodes which enhance its quality.

This leads naturally in 2D and 3D to anisotropic grids which are able to achieve a reasonable trade-off between accuracy and efficiency. Anisotropic grids are often used in the context of finite element and finite volume methods and thus appear very frequently in practical applications [13, 15, 12]. However, how to obtain such an anisotropic mesh for a given application is a very open problem. Unfortunately, not many numerical analysts focus on the art of designing precise grids but rather on discretization techniques for the solution of the continuous problem (e.g. the governing differential equation).

In this paper, we study how to automatically obtain curvature-adapted surface meshes from low-quality ones. The initial mesh may be too coarse in certain regions or too dense in others. Also, in practice one also has to deal with not properly oriented triangles. While a strechted triangle is not per se unacceptable, it should not be elongated in the direction where the curvature varies a lot. Figure 1 shows what our method is capable of. The original mesh on the left is extremely coarse and uniform. The adapted mesh, on the other hand, is denser and refined where necessary. The stretched triangles reflect nicely the curvature. For example, the curvature along the dashed arrows varies more than along the corresponding solid ones. Hence, we need to refine more in the

direction of the dashed arrows which means that the triangles will be stretched along the solid arrows.

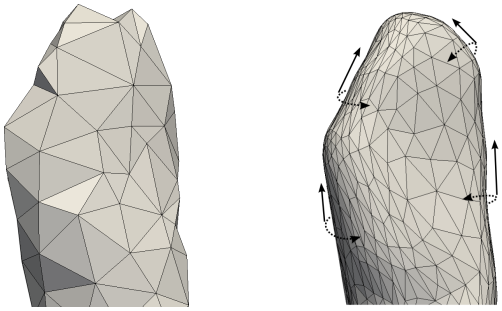


Figure 1: A coarse input mesh on the left and the adapted mesh using the novel adaptation procedure on the right. Both figures show a bone from the vertebrae mesh in Subsection 5.4.

The key idea of our method is to combine surface approximation by radial basis functions with the higher dimensional embedding technique. Radial basis functions (RBFs) have been used for decades in the context of multivariate data approximation [10, 20, 26, 9]. Ironically, their main selling point is that they can be used to interpolate unstructured data without relying on a mesh. Even though they are most commonly used in the context of *meshfree* data approximation, they sometimes have been employed in the context of mesh repair. Carr et al. used polyharmonic radial basis functions to fill in incomplete meshes. Similarly, Marchandise et al. [22] developed a method to repair meshes obtained from a CAD model or an STL triangulation [4].

The higher dimensional embedding (HDE) was introduced by Cañas et al. in [2]. Since then several authors have expanded their ideas [21, 5, 3]. HDE produces an anisotropic triangular curvature-adapted surface mesh that fits an input surface. The anisotropy is obtained by finding a higher-dimensional space in which the mesh is assumed to be uniform and isotropic. Previously other approaches based on metric tensor fields [16, 14] or minimizing objective functionals [24, 8] have been studied. However, in order to be able to apply both of these strategies one needs to have a priori or a posteriori knowledge of the error, which depends on the problem itself. The HDE, on the other hand, does not require any information on the error. It is solely based on information provided by the embedding map.

This paper is organized as follows. After introducing the higher dimensional embedding technique and radial basis functions in the second and third sections, we present our novel surface remeshing approach in the fourth and supplement it in the fifth section with extensive numerical studies and successfully apply it to real-life problems.

For the rest of the paper, we will assume that the surface Γ is given implicitly by the zero level set of some function $F\colon \Omega \subseteq \mathbb{R}^3 \to \mathbb{R}$, i. e.

$$\Gamma = \left\{ (x, y, z)^T \in \Omega \mid F(x, y, z) = 0 \right\} , \tag{1}$$

for some bounded domain Ω.

2 Higher Dimensional Embedding

The higher dimension embedding technique fits a triangular surface mesh to a given geometry by enlarging the space we are originally interested in. The key assumption of HDE is that a uniform isotropic mesh in a higher-dimensional space will correspond to an anisotropic mesh in a lower-dimensional space.

This concept is best explained with Figure 2. The left image shows a uniform isotropic triangular mesh in $\mathbb{R}^3$. However, if projected onto $\mathbb{R}^2$ the mesh becomes anisotropic which is shown in the picture on the right.

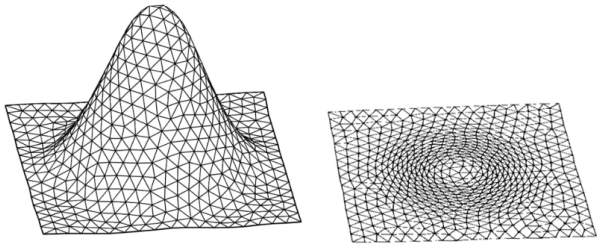


Figure 2: An isotropic mesh in the higher-dimensional space $\mathbb{R}^3$ (left) and the corresponding anisotropic mesh in $\mathbb{R}^2$ (right). This picture encapsulates the key idea behind the higher dimensional embedding technique [21].

To obtain an anisotropic curvature-adapted mesh of an input surface $\Gamma \subset \mathbb{R}^3$, we define the embedding map $\Psi : \Gamma \to \mathbb{R}^6$ by

$$\Psi(\mathbf{x}) = (x,\, y,\, z,\, \sigma\, n_x,\, \sigma\, n_y,\, \sigma\, n_z)^T\,, \tag{2}$$

where $\mathbf{n} = (n_x, n_y, n_z)^T$ denotes the unit normal to Γ at $\mathbf{x} = (x,\, y,\, z)^T$ and $\sigma > 0$ is a constant that controls the influence of the normals in the embedding map. For small σ the mapping Ψ is close to the identity in $\mathbb{R}^3$ embedded in $\mathbb{R}^6$, i. e. $\Psi(\mathbf{x}) \approx (x, y, z, 0, 0, 0)^T$. Larger values of σ put more emphasis on the normal components of the surface Γ.

In the higher-dimensional space points on Γ are enriched with surface normal information. Consider for instance two generic points $\mathbf{a}, \mathbf{b} \in \Gamma$ and the edge $\overline{\mathbf{ab}}$. If the surface is flat, the normals at the endpoints are the same. Consequently, the length of the edge in $\mathbb{R}^3$, denoted with $l_{\mathbf{ab}}$, coincides exactly with the length measured in the higher-dimensional space $l^{6d}_{\mathbf{ab}}$. On the other hand, if the surface is curved, the normals at the endpoints are different and $l^{6d}_{\mathbf{ab}}$ becomes much larger than $l_{\mathbf{ab}}$.

2.1 Mesh Generation in Higher Dimensions

There are two established ways to build a uniform isotropic mesh in the higher-dimensional space. In [21], the authors propose an approach based on the Restricted Centroidal Voronoi Tessellation. Starting from the embedding defined in (2), they compute the Voronoi Tessellation of an initial point set $\{\mathbf{p}_i\}_{i=0}^N$. Then, they optimize this tessellation to get a Restricted Centroidal Voronoi Tessellation, i. e. a Restricted Voronoi Tessellation where $\mathbf{p}_i = \mathbf{g}_i$ for all $i = 1, 2, \ldots N$. Here, $\mathbf{g}_i$ denotes the centroid of the Voronoi cell associated with the point $\mathbf{p}_i$. The triangular mesh is finally obtained by duality from this Voronoi diagram.

This mesh generation strategy presents a couple of challenges. Firstly, the computation and the optimization of the Restricted Voronoi Tessellation is made in the higher-dimensional space, so it incurs a high computational cost. Secondly, the sharp features of the input geometry are smoothed and over refined. Moreover this method can produce invalid mesh with self intersection.

In [3, 5], a different method was proposed to construct a uniform isotropic mesh in the higher-dimensional space. The idea is to exploit the standard scalar product in the higher-dimensional space. Consider three points $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ on the surface Γ, then the lengths and the angles in the higher-dimensional space are defined via

$$\begin{aligned} l^{6d}_{\mathbf{ab}} &:= \big|\big|\, \Psi(\mathbf{a}) - \Psi(\mathbf{b}) \,\big|\big|_{6\mathrm{d}} := \sqrt{\Big(\Psi(\mathbf{a}) - \Psi(\mathbf{b}), \Psi(\mathbf{a}) - \Psi(\mathbf{b})\Big)_{6\mathrm{d}}}\,, \\ \cos\left(\theta^{6d}_{\mathbf{acb}}\right) &:= \frac{\Big(\Psi(\mathbf{a}) - \Psi(\mathbf{c}), \Psi(\mathbf{b}) - \Psi(\mathbf{c})\Big)_{6\mathrm{d}}}{l^{6d}_{\mathbf{ac}}\, l^{6d}_{\mathbf{bc}}}\,. \end{aligned} \tag{3}$$

Now, one can fix a target edge length l^{6d} in the higher-dimensional space and modify an initial mesh Γ_h^{ini} in such a way that

$$l^{6d}_{\mathbf{e}} \approx l^{6d} \qquad \text{and} \qquad \cos\left(\theta^{6d}_{\alpha}\right) \approx \frac{1}{2} \qquad \forall \mathbf{e} \in \mathcal{E} \quad \text{and} \quad \forall \alpha \in \mathcal{A}\,, \tag{4}$$

where $\mathcal{E}$ and $\mathcal{A}$ are the sets of edges and angles of all mesh triangles.

Remark 1 *The method proposed in [3, 5] does not embed the mesh in $\mathbb{R}^6$. Only the lengths and angles are computed in the higher-dimensional space. The mesh is modified in $\mathbb{R}^3$ by standard mesh operations for triangular elements such as edge flipping, edge splitting/contraction and node smoothing.*

This method overcomes all of the previously discussed drawbacks. It is computationally cheaper, the sharp features of the input mesh are preserved and no triangle of the resulting mesh intersects with itself [5].

3 Geometry Reconstruction with Radial Basis Functions

Radial basis functions are commonly divided into two categories: positive definite and conditionally positive definite functions. We state both definitions here.

Definition 1 (Positive definite function) *Let $\Phi\colon \mathbb{R}^d \to \mathbb{R}$ be a continuous function. We define the matrix $A_{\Phi,X}$ via its ij^{th} entry*

$$a_{ij} = \Phi(\mathbf{x}_i - \mathbf{x}_j)\,,$$

for any data set $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^d$ of arbitary length $N \geq 1$. The function Φ is called positive definite *if the quadratic form*

$$\mathbf{c}^T A_{\Phi,X} \mathbf{c} \tag{5}$$

is positive for all vectors $\mathbf{c} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$.

Definition 2 (Conditionally positive definite function) *Let $\mathcal{P}_m(\mathbb{R}^d)$ denote the space of d variate polynomials with absolute degree at most m and dimension $q := \dim \mathcal{P}_m(\mathbb{R}^d) = \binom{m-1+d}{d}$. For a basis $p_1, \dots, p_q$ of this space, define the $N \times q$ polynomial matrix P_X through its ij^{th} entry*

$$p_{ij} = p_i(\mathbf{x}_j)\,,$$

where $\mathbf{x}_j \in X$. The function Φ is called conditionally positive definite of order m *if the quadratic form (5) is positive for all X and for all $\mathbf{c} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$ which additionally satisfy the constraint $P_X^T \mathbf{c} = \mathbf{0}$.*

One typically speaks of radial basis functions if one additionally assumes that Φ is a radial function, i.e. there exists a function $\phi\colon \mathbb{R}_{\geq 0} \to \mathbb{R}$ such that $\Phi(\mathbf{x}) = \phi(\|\mathbf{x}\|)$. Trivially, a positive definite function is also a conditionally positive definite function of order $m = 0$ and conditionally positive functions of order m are also conditionally positive for any order higher than m. Hence, the order usually shall denote the smallest positive integer m.

Suppose we want to recover a function $f\colon \mathbb{R}^d \to \mathbb{R}$ known on some data set $X = \{\mathbf{x}_i\}_{i=1}^N$. We can solve the interpolation problem

$$f(\mathbf{x}_i) = s(\mathbf{x}_i), \qquad 1 \leq i \leq N \tag{6}$$

for the interpolant $s\colon \mathbb{R}^d \to \mathbb{R}$ using radial basis functions by making the ansatz

$$s(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \Phi(\mathbf{x} - \mathbf{x}_j) \tag{7}$$

in the case of a positive definite Φ and

$$s(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \Phi(\mathbf{x} - \mathbf{x}_j) + \sum_{k=1}^{q} \beta_k p_k(\mathbf{x}) \tag{8}$$

in the case of conditionally positive definite functions. The coefficients $\boldsymbol{\alpha} = (\alpha_j) \in \mathbb{R}^N, \boldsymbol{\beta} = (\beta_k) \in \mathbb{R}^q$ need to be determined by applying the interpolation condition (6) to either (7) or (8). Hence one needs to solve either

$$A_{\Phi,X}\boldsymbol{\alpha} = \mathbf{f} \qquad \text{or} \qquad \begin{pmatrix} A_{\Phi,X} & P_X \\ P_X^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}, \tag{9}$$

where $\mathbf{f} = (f(\mathbf{x}_j))$.

For positive definite functions, the linear system is positive definite by construction. Hence the coefficients can be determined uniquely. It is also not difficult to verify that the second choice for an interpolant leads to unique coefficients in the case of conditionally positive definite functions, see [26, Theorem 8.21] for details. In the case of conditionally positive definite functions the Courant-Fischer theorem guarantees that at least $N - q$ eigenvalues of the matrix $A_{\Phi,X}$ are positive.

Another criterion to classify RBFs is whether they have compact support or not. This is an advantageous feature for very large data sets since the matrix $A_{\Phi,X}$ becomes sparse if the support radius is small enough. It is well known that for compactly supported RBFs the polynomial part in (8) has to vanish. There

Global support		Compact support ($d = 3$)	
e^{-r^2}	Gaussian	$\phi_{3,1}(r) = (1-r)_+^4(4r+1)$	C_2
$\sqrt{1+r^2}$	Multiquadric	$\phi_{3,2}(r) = (1-r)_+^6(35r^2+18r+3)$	C_4
$1/\sqrt{1+r^2}$	Inv. Multiquadric	$\phi_{3,3}(r) = (1-r)_+^8(32r^3+25r^2+8r+1)$	C_6
r^3	Polyharm. Spline	due to Wendland [25]	

Table 1: Common examples of globally and compactly supported RBFs. All functions are positive definite except the polyharmonic spline which is conditionally positive definite for $d = 3$ with (minimal) order 2. For positive r the truncation operator $(\cdot)_+$ leaves its argument unaltered. For negative arguments it is set to zero. The last row indicates the regularity of the Wendland RBFs.

are no nontrivial conditionally positive definite functions with compact support [26, Theorem 9.1]. Wendland [25] presented a class of compactly supported radial basis functions which consist of polynomials within their support. The degree of the polynomials for a given space dimension and smoothness parameter is minimal.

Common examples of RBFs are shown in Table 1.

3.1 Surface reconstruction with RBFs

We cannot simply replace the target function f in (6) with the function F whose zero level set describes the implicit surface (1) since the right hand sides of the linear systems (9) vanish which implies that the coefficients vanish as well. Carr et al. therefore made the additional assumption that the normal vectors are known. One then can also prescribe on-surface and off-surface points. Assume that the points on the surface are denoted with $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and the corresponding normal vectors with $M = \{\mathbf{n}_1, \ldots, \mathbf{n}_N\}$. Then one can define the surface interpolation problem

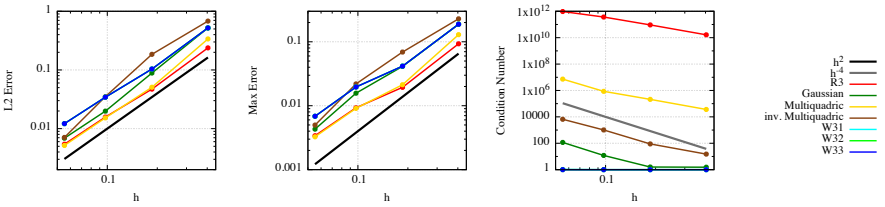
$$\begin{aligned} s(\mathbf{x}_i) = F(\mathbf{x}_i) &= 0, & 1 \le i \le N \quad & \text{(on-surface points)} \\ s(\mathbf{x}_i + \varepsilon\mathbf{n}_i) = F(\mathbf{x}_i + \varepsilon\mathbf{n}_i) &= \varepsilon, & N+1 \le i \le 2N \quad & \text{(off-surface points)} \end{aligned} \tag{10}$$

for some parameter $\varepsilon > 0$. Since the right hand side is no longer zero, we find now nontrivial solutions to the linear systems (9). Actually, it is enough to define just one off-set point to get a nontrivial solution. Moreover it is even possible to add more constraints. One possibility is to consider $s(\mathbf{x}_i - \varepsilon\mathbf{n}_i) = F(\mathbf{x}_i - \varepsilon\mathbf{n}_i) = -\varepsilon$ for $2N + 1 \le i \le 3N$.

3.2 Partitioning the RBF Interpolant

To reduce the computational cost when solving the linear system (9) or when projecting points onto the surface (see Subsection 4.3) we consider a disjoint partition of the surface

$$\Gamma_h^{ini} = \Gamma_{1,h}^{ini} \cup \Gamma_{2,h}^{ini} \cup \ldots \cup \Gamma_{K,h}^{ini} \tag{11}$$



(a) L_2 error (b) L_∞ error (c) Condition numbers (d) Legend

Figure 3: Comparision of different RBFs for varying mesh norms for $\varepsilon = 1/8$ and the implicit function F given by (13). The arguments are scaled by 1/10.

for $K \in \mathbb{N}$. Let X_i and M_i denote the points on the surface patch $\Gamma_{i,h}^{ini}$ and the corresponding normals. Now we can solve the interpolation problem (10) locally defined on each patch $\Gamma_{i,h}^{ini}$ and obtain local interpolants s_i whose zero level sets $\Gamma_0^{s_i}$ approximate $\Gamma_{i,h}^{ini}$.

Then the global interpolant is defined as

$$s(x,y,z) := \begin{cases} s_1(x,y,z) & \text{if } (x,y,z)^T \in \Gamma_{1,h}^{ini} \\ s_2(x,y,z) & \text{if } (x,y,z)^T \in \Gamma_{2,h}^{ini} \\ \vdots & \vdots \\ s_K(x,y,z) & \text{if } (x,y,z)^T \in \Gamma_{K,h}^{ini} \end{cases} . \tag{12}$$

3.3 Errors and conditioning of the RBF interpolant

In order to study the approximation quality and the conditioning of the RBF interpolant, we choose a simple ellipsoid surface which is given by the zero level set of the function

$$F(x,y,z) = x^2/4 + y^2/4 + z^2/0.2 - 1. \tag{13}$$

The reason for this relatively simple surface is that it allows to compute the error between our approximation and the exact solution. For nested point sets on the ellipsoid where the mesh norm h (the radius of the largest ball which does not contain any data point) is reduced each time by a factor of two, we see the results in Figure 3. The L_2 and L_∞ errors of all radial basis function approximations considered seem to converge at least quadratically.

The condition numbers, however, behave quite differently. Whereas all globally supported radial basis functions behave eventually roughly like h^{-4}, the condition numbers of the compactly supported radial basis functions grow only very mildly (which is not even visible in the log plot). As mentioned before, this is not a surprise as their compact support leads to a sparse matrix system. Even though these results might not yet capture the long term behavior, they are from a computational point of view more interesting as one would like to use as few points as possible.

We report here that other surfaces (sphere and one-sheeted hyperboloid) have been examined with similar results: at least quadratic convergence for both errors and a growth in condition number of at least h^{-4}.

4 A Novel Anisotropic Surface Remeshing Approach

In this section, we describe in detail the novel anisotropic curvature adapted remeshing procedure. Starting from a conformal triangular surface mesh Γ_h^{ini}, we proceed as follows:

(1) Build RBF approximation Γ_0^s of the initial surface mesh Γ_h^{ini} (Section 4.1) and

(2) Construct final adapted mesh Γ_h^{fin} via HDE including:

- several local mesh modifications (Section 4.2) and
- projections onto the surface Γ_0^s (Section 4.3)

4.1 Approximation of Γ_0^s

When constructing the continuous approximation of Γ_h^{ini}, the user has to specify the type of radial basis function Φ, the parameter ε, the number of partitions K and the overlap parameter Ov. In order to setup the interpolation problem defined in (10), we use the vertices of Γ_h^{ini} as interpolation nodes X. By averaging the normals of the triangles that share a generic vertex $\mathbf{x}_i$, we define the normals $\mathbf{n}_i$ at this point and obtain the set M. Finally, we construct and solve the linear system (9) to derive the continuous interpolant s.

4.2 Remeshing Algorithms

Next, using Algorithm 1 we modify Γ_h^{ini} so that it becomes a uniform isotropic mesh in the higher-dimensional space. The algorithm takes two inputs: the parameter σ, appearing in the embedding map Ψ, as well as the target length in the higher-dimensional space l^{6d}. It employs several mesh modifiction procedure such as edge flipping, edge contraction, edge splitting and vertex smoothing [19, 18, 6, 16], computing only the edge lengths and angles in the higher-dimensional space. We explain these mesh modificatons in the following paragraphs.

Edge-Flip. Consider two triangles $\Delta_{\mathbf{abc}}$ and $\Delta_{\mathbf{bad}}$ whose common edge is $\overline{\mathbf{ab}}$. Flipping the edge $\overline{\mathbf{ab}}$ creates two new triangles $\Delta_{\mathbf{cda}}$ and $\Delta_{\mathbf{dcb}}$ and replaces the edge $\overline{\mathbf{ab}}$ with $\overline{\mathbf{cd}}$, see Figure 4.

It is not always possible to flip the edge $\overline{\mathbf{ab}}$ in a triangular surface mesh. The edge $\overline{\mathbf{ab}}$ is flippable if it satisfies the following conditions:

(i) $\overline{\mathbf{ab}}$ does not belong to the boundary of the mesh;

(ii) $\overline{\mathbf{cd}}$ does not already belong to the mesh;

(iii) the internal angles of the triangles satisfy

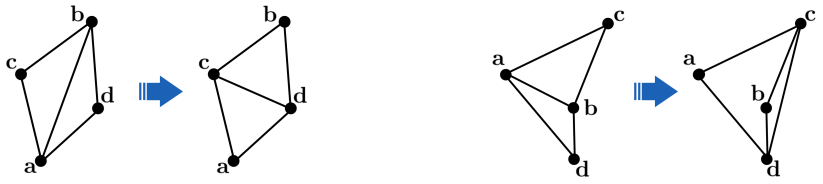
$$\theta_{\mathbf{abc}} + \theta_{\mathbf{abd}} < \pi \qquad \text{and} \qquad \theta_{\mathbf{bac}} + \theta_{\mathbf{bad}} < \pi\,;$$

(iv) the angle between the normals of the triangles $\Delta_{\mathbf{abc}}$ and $\Delta_{\mathbf{bad}}$ is lower than some threshold value. Here we consider $80°$.

Algorithm 1 The anisotropic mesh adaptation.

IMPROVE(Γ_h^{ini}, `maxIter`)

for $i \in \{1, ..., \texttt{maxIter}\}$ **do**
  repeat
    contract all edges which satisfy $l_{\mathbf{e}}^{6d} < 0.5\, l^{6d}$;
    smooth 30% of vertices;
    flip edges if Equation (14) holds;
  until $l_{\mathbf{e}}^{6d} \geq 0.5\, l^{6d}$ for all edges $\mathbf{e}$
  split all edges which satisfy $l_{\mathbf{e}}^{6d} > 1.5\, l^{6d}$;
  flip edges if Equation (14) holds;
  smooth 30% of vertices;
  flip edges if Equation (14) holds;
end for



(a) (b)

Figure 4: Example for an edge flip (a) and two neighboring triangles in (b) that do not satisfy criterion (iii).

If an edge $\overline{\mathbf{ab}}$ satisfies all these conditions, it is flippable. We dedice to actually flip the edge $\overline{\mathbf{ab}}$ if the relation

$$\theta_{\mathbf{acb}}^{6d} + \theta_{\mathbf{adb}}^{6d} > \pi \tag{14}$$

holds, where $\theta_{\mathbf{acb}}^{6d}$ and $\theta_{\mathbf{adb}}^{6d}$ are the angles measured in the higher-dimensional space. This condition is inspired by the well-known Delaunay criterion, reinterpreted in the higher-dimensional space [7].

Edge contraction and edge splitting. We try to approximately match all edge lengths of our mesh to a given target edge length l^{6d} by

- contracting the edge $\mathbf{e}$ if $l_{\mathbf{e}}^{6d} < 0.5\, l^{6d}$,
- splitting the edge $\mathbf{e}$ if $l_{\mathbf{e}}^{6d} > 1.5\, l^{6d}$,
- and otherwise applying neither edge contraction nor edge splitting.

Node Smoothing. Unlike the previous mesh modification procedures, this operation does not change the topology of the mesh. It simply moves a point $\mathbf{x}$ to a new location $\mathbf{x}'$. This new location can be computed as follows:

$$\mathbf{x}' = \mathbf{x} + \alpha \sum_{\mathbf{x}_i \in \omega_{\mathbf{x}}} w(d(\mathbf{x}, \mathbf{x}_i))\mathbf{u}_i\,. \tag{15}$$

Here α is a scaling parameter, $w : \mathbb{R} \to \mathbb{R}$ is a weight function, $\omega_{\mathbf{x}}$ is the set of vertices that are connected to the old point $\mathbf{x}$, $\mathbf{u}_i$ denote the unit vectors that point from $\mathbf{x}$ to $\mathbf{x}_i$ and d is the distance between $\mathbf{x}$ and $\mathbf{x}_i$.

Different choices of α and w in (15) lead to different node smoothing methods. Here we use the node smoothing method proposed in [5]. That means, the distances d are computed in the higher-dimensional space and the parameter α is set to $1/(\sum_{\mathbf{x}_i \in \omega_{\mathbf{x}}} w(d(\mathbf{x}, \mathbf{x}_i)))$ and $w(x) := (1 - x^4)e^{-x^4}$ as introduced in[1].

4.3 Projection of nodes onto Γ_0^s

In this section, we describe how we apply edge splitting and edge contraction while constraining all new points onto the RBF surface Γ_0^s. Applying these operations naively will fail.

Projection after Edge Splitting or Node Smoothing

If we apply edge splitting or node smoothing on a triangular surface mesh, it is not a priori guaranteed that these operations yield a new point that lies on the reconstructed surface Γ_0^s.

Consider, for instance, the edge splitting operation. If we simply halve the edge, the new point lies only under very special circumstances on Γ_0^s. Moreover, since the new point is usually not on the surface, the unit normal is not defined. Node smoothing leads to a similar problem.

To avoid this issue, we exploit a projection algorithm. If we split an edge or move a point using node smoothing, we project the resulting point onto the RBF surface reconstruction Γ_0^s. Hartmann [17] provides a robust algorithm to project a point onto a surface defined via the zero level set of a function. This procedure is a combination of evaluating foot points on tangent planes and approximating foot points on tangent parabolas. It requires only first order derivatives and uses a steepest descent method. In our case, the surface is described via Γ_0^s, i. e. the zero level set of the interpolant s. Figure 5 provides a two dimensional example of node projection after edge splitting.

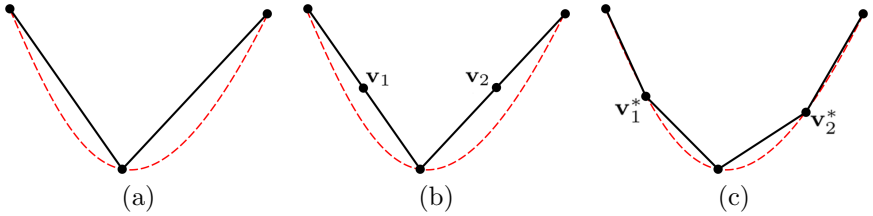


Figure 5: Figure (a) shows a coarse input mesh (solid line) which approximates the reconstructed curve (dashed line). In Figure (b) the edges are halved. The midpoints $\mathbf{v}_1$ and $\mathbf{v}_2$ do not lie on the reconstructed curve. Finally, Figure (c) shows how the points are projected onto the dashed curve using a steepest descent method which yields new points $\mathbf{v}_1^*$ and $\mathbf{v}_2^*$.

Edge Contraction to End Point

An edge $\mathbf{e}$ can be contracted to different locations. We decide to contract it to one of the endpoints [7] to avoid having to project this point onto Γ_0^s. The endpoints lie on Γ_0^s by construction where the normals are well-defined. Figure 6 provides a two-dimensional example.

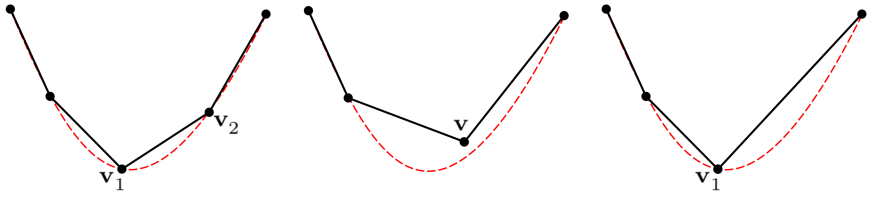


Figure 6: Figure (a) shows a fine input mesh (solid line) which approximates the reconstructed curve (dashed line). In Figure (b), the edge $\overline{\mathbf{v}_1\mathbf{v}_2}$ is contracted to its midpoint $\mathbf{v}$, which does not lie on the reconstructed curve. However, in Figure (c) the edge $\overline{\mathbf{v}_1\mathbf{v}_2}$ in contracted to the end point $\mathbf{v}_1$ which by construction automatically lies on the dashed curve.

Defining normals on Γ_0^s

Consider the interpolant s and the RBF approximation of the discrete input surface Γ_0^s. We use the gradient components of the function s to obtain the unit normal of Γ_0^s at a point $\mathbf{x}$.

Let $(\nabla s)_x$, $(\nabla s)_y$ and $(\nabla s)_z$ be the x, y and z the components of s, respectively. Then, the unit normal $\mathbf{n} = (n_x, n_y, n_z)^T$ at point $\mathbf{x}$ is defined as

$$n_x := \frac{(\nabla s)_x}{||\nabla s||}, \quad n_y := \frac{(\nabla s)_y}{||\nabla s||}, \quad n_z := \frac{(\nabla s)_z}{||\nabla s||}, \tag{16}$$

where $||\nabla s||$ denotes the standard Euclidean norm of ∇s. We point out that by construction $||\nabla s||$ does not vanish on Γ_0^s.

Overlapping and Projection

To reduce the computational effort, we may choose to partition the domain according to (11). Then the interpolant s whose zero level set defines the surface Γ_0^s is given by (12).

Each time we project some node onto the surface, we have to determine the local function s_i on which we want to project. However, near the boundary between patches the node can be projected on more than one local interpolant. Also having to determine which local interpolant to project on involves a costly point search algorithm. Unfortunately, we cannot use a standard nearest-neighbor search since it is not a priori guaranteed that the closest reconstructed patch is the one we want to project our point on.

For a given patch $\Gamma_{i,h}^{ini}$ with $1 \leq i \leq K$, we assign to each of its points the flag i. Whenever we halve an edge, we have to project the new point on one of the patches. If both endpoints belong to the same patch, the new point

inherits the same flag and we project it on this patch. On the other hand, if the endpoints have different flags, say i and j, we assign to the new point the flag $k := \min\{i, j\}$ and project it on $\Gamma_0^{s_k}$.

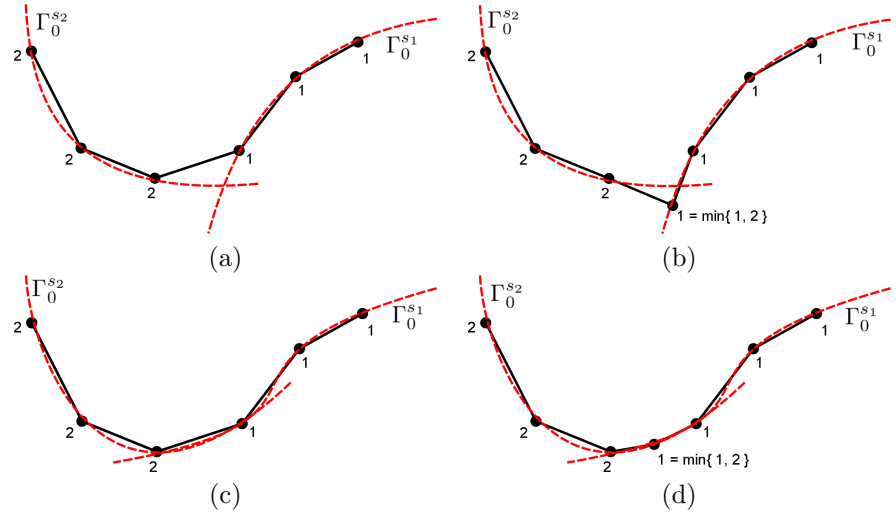


Figure 7: A two-dimensional example illustrating the discontinuity arising from projecting on a partitioned surface. The dashed lines are the reconstructed curves. For each point we denote the patch it belongs to by a number (a). When we split an edge whose endpoints belong to different patches, we choose the smaller one, leading to a discontinuity (a kink) in the surface (b). The third figure shows the new curves constructed from overlapping interpolation point sets (c). This considerably reduces the discontinuity along the boundary between both patches (d).

This strategy will avoid the onerous searching procedure, but it may lead to discontinuities near the boundary of two patches, see Figure 7 (a) and (b). Hence, we introduce overlapping patches. Consider a partition $\Gamma_{i,h}^{ini}$ and the set of vertices that lie on its boundary. We add to the sets X_i and M_i the vertices of Γ_h^{ini} which share a triangle T that has at least one vertex on the boundary of the patch $\partial\Gamma_{i,h}^{ini}$, see Figure 8. This process is usually also done for the neighboring patch, creating two patches which now overlap. If necessary, we successively increase the overlap thus enlarging the sets X_i and M_i, see Figure 8 (c) and (d). The parameter Ov will count how many overlap layers have been added. The impact of this parameter on the global approximation is studied in Subsection 5.1.1.

5 Numerical Examples

In the following, we study different input meshes to assess the quality and robustness of our new anisotropic remeshing method, described in Section 4. We start by examining three model geometries to analyze the impact of the different parameters such as the overlap parameter Ov, the edge length l^{6d} and the embedding parameter σ on the final mesh Γ_h^{fin}. Additionally, we show that

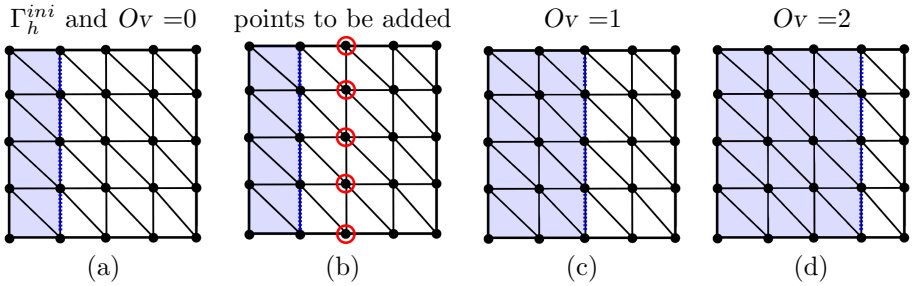


Figure 8: A simple example of the overlap for a structured mesh on a square. In (a) we show the patch $\Gamma_{1,h}^{ini}$. We show in (b) the patch $\Gamma_{1,h}^{ini}$, its boundary $\partial\Gamma_{1,h}^{ini}$ and the points which will be added are highlighted with circles. The cases $Ov = 1$ and $Ov = 2$ are provided in (c) and (d), respectively.

our method can not only be used to fill in more detail but also to thin out out very complex meshes. In the last subsection, we apply our method to real-life examples from medical applications for which we would like to build anisotropic meshes.

To assess the degree of anisotropy of our mesh, we compute the *global aspect ratio*

$$q_{\Gamma_h} := \max_{T \in \Gamma_h} q_T \,, \tag{17}$$

where $q_T := R_T/(2\, r_T)$ is the so called *aspect ratio* of the triangle T. Here R_T and r_T are the radii of the circumscribed and inscribed circle of T, respectively [23]. We observe that $q_T \geq 1$ by construction. If $q_T = 1$, then T is an equilateral triangle, while if $q_T \gg 1$, the triangle T is stretched.

5.1 Sensitivity Analysis of the Parameters

We analyze the most relevant parameters in the adaptation procedure. More precisely, in Subsection 5.1.1, we focus on the influence of the Ov parameter when constructing Γ_0^s. In Subsection 5.1.2, we conduct a sensitivity analysis of the edge length l^{6d} and the embedding parameter σ. Throughout this subsection, we consider the discrete input surface shown in Figure 9 (left).

5.1.1 Overlap Analysis

We study more carefully the influence of the overlap parameter Ov. This is a key parameter when dealing with a partition (11) since it reduce the discontinuities of Γ_0^s along the patch boundaries. To understand its effect on the reconstructed geometry, we use thin plate splines, $\varepsilon = 0.1$ and $K = 20$.

From Figure 9 we infer that the smoothness of the approximation increases for higher values of the parameter Ov. Heuristically, an overlap parameter of 2 or 3 yields acceptable approximations.

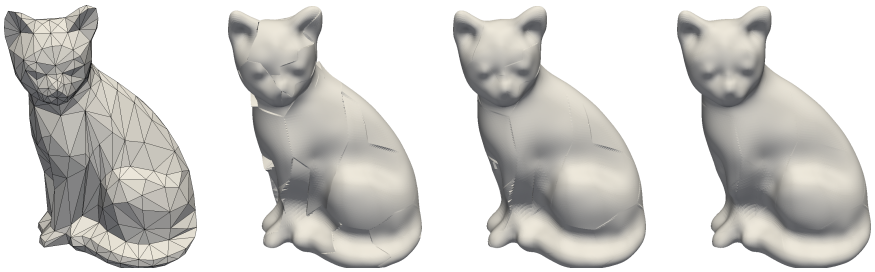


Figure 9: The cat geometry used for the sensitivity analysis of the overlap parameter Ov, the edge length l^{6d} and the embedding parameter σ. The RBF surface reconstruction Γ_0^s for the cat with different values for Ov.

5.1.2 Mesh Adaptation Parameters

Next, we give a sensitivity analysis of the target length l^{6d} and the embedding parameter σ. To construct Γ_0^s, we use as before thin plate splines, $\varepsilon = 0.1$ and $K = 20$. Additionally, we set $Ov = 2$ and consider the following sets of values

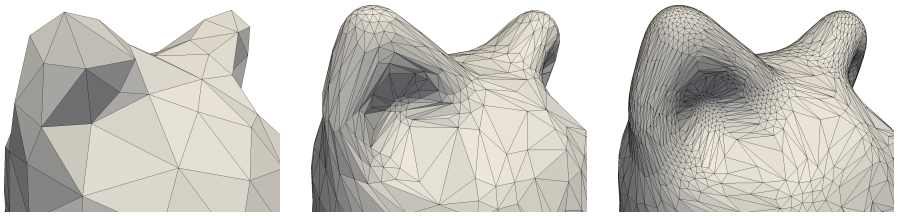
$$\sigma \in \{1,\, 5,\, 10\} \quad \text{and} \quad l^{6d} \in \{20,\, 15,\, 10\}\,.$$

In Figure 11, we show the resulting meshes and provide a quantitative analysis in Table 2. Our remeshing algorithm behaves as expected. If we fix the parameter σ and decrease l^{6d}, the resulting mesh becomes finer, see the columns in Figure 11 and Table 2.

On the other hand, if we fix the target edge length l^{6d} and increase the parameter σ, we observe a similar behavior, see the rows of Figure 11 and Table 2, respectively. The lengths in the higher-dimensional space (3) grow when increasing the parameter σ, which implies there is a higher sampling to achieve the same target embedded length. However, since σ controls the influence of the normals in the embedding map, the sampling is localized only in high curvature regions, see the details in Figure 10.

$l^{6d} \setminus \sigma$	1		5		10	
20	q_{Γ_h}	2.597e+00	q_{Γ_h}	1.449e+01	q_{Γ_h}	3.567e+01
	#ele.	1826	#ele.	3318	#ele.	7743
15	q_{Γ_h}	1.684e+00	q_{Γ_h}	8.441e+00	q_{Γ_h}	2.076e+01
	#ele.	2142	#ele.	4960	#ele.	13 248
10	q_{Γ_h}	2.419e+00	q_{Γ_h}	1.914e+01	q_{Γ_h}	2.935e+01
	#ele.	3264	#ele.	10 029	#ele.	27 943

Table 2: Number of elements and global aspect ratios q_{Γ_h} for different values of σ and l^{6d}.



$\sigma = 1$ $\sigma = 5$ $\sigma = 10$

Figure 10: A detail of the cat for $l^{6d} = 15$ and different values of σ. In this region the normals vary greatly. Hence, the mesh adapation procedure refines more and more for increasing values of σ.

5.2 Bunny

We examine the well-known Stanford bunny, depicted in Figure 12. To construct Γ_0^s, we use thin plate splines, $\varepsilon = 0.001$, $K = 200$, $Ov = 3$. We run the anisotropic adaptation procedure with $l^{6d} = 0.001$ and $\sigma = 1.0$. In Figure 12, we show both the initial and the resulting mesh. The resulting mesh is strongly anisotropic, in fact the global aspect ratio q_{Γ_h} is $3.989e{+}02$ and it is more refined than the initial one. While the initial mesh has 69 451 elements, the final one has 110 350 triangles.

The triangles are aligned and stretched according to the curvature of the surface. This can be clearly seen when zooming in on the ears of the bunny, see Figures 13. In the initial mesh Γ_h^{ini}, the triangles are too big to capture the helix of the bunny's ears and their orientation and shape do not reflect the curvature of the surface. However in the final mesh Γ_h^{fin} all these features disappear.

Moreover, the final anisotropic adapted mesh makes details in the mesh more apparent. For instance, unlike for the initial mesh Figure 12 (left), finer details in the fur of the bunny are clearly visible in the final mesh, see Figure 12 (right).

5.3 Lucy

In both previous examples, we started from a coarse initial mesh that poorly approximates the input geometry. However, our mesh anisotropic mesh adaptation procedure can also be used to simplify a dense initial mesh, coarsening it in regions with little curvature variation.

We consider the geometry shown in Figure 14. This mesh is another well-known benchmark in surface mesh adaptation ("Lucy"). Here the mesh is so fine that one would not be able to see the faces of the triangles because of the density of the edges, see Figure 14 left. Hence we only show the faces of the surface without the edges. To construct the surface approximation, we use thin plate splines, $\varepsilon = 0.1$, $K = 3000$, $Ov = 3$ and the $\sigma = 10$. We point out that the number of partitions is rather high. If we want to reduce the number of partitions or avoid dealing with large dense linear systems, we can employ compactly supported RBFs.

The initial mesh offers a very fine approximation of the input geometry, but managing this huge data set requires an unacceptably high computational effort. By choosing a large target length l^{6d} our anisotropic adaptation procedure effectively becomes a mesh simplification method. The detail in Figure 15 show

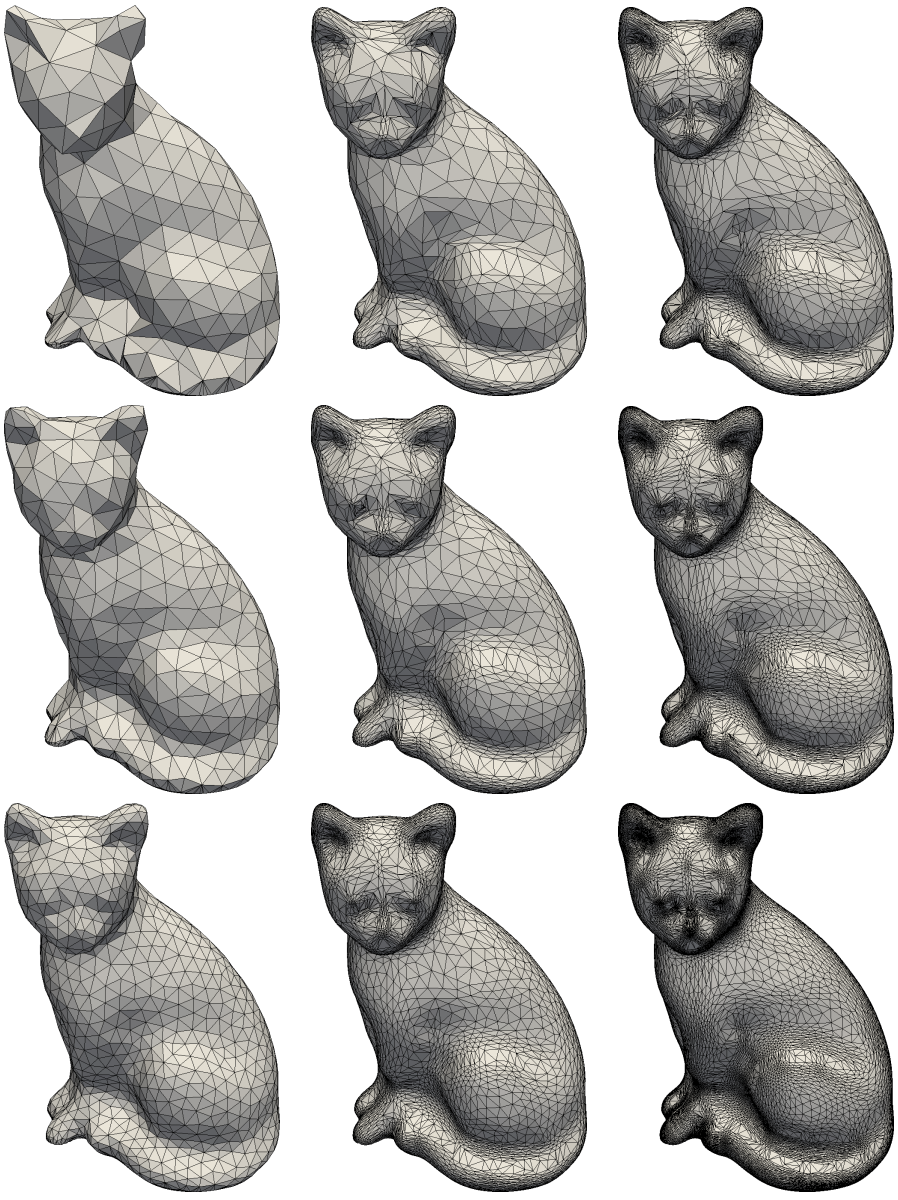
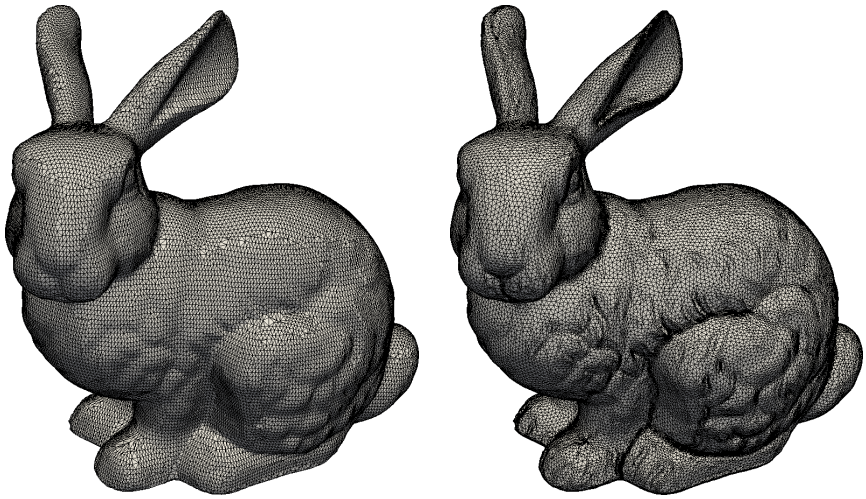


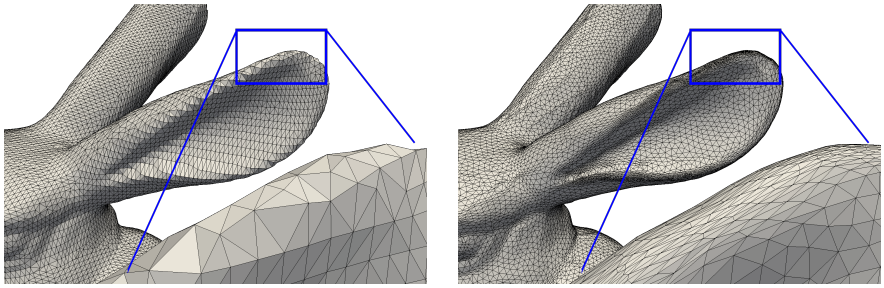
Figure 11: Adapted cat meshes for different values of σ (columns) and $l^{6d} = 20, 15, 10$ (rows, from top to bottom).

that the shape of the initial mesh is still preserved, however, using considerably fewer elements. To be more precise, we reduce the number of elements by 73% for $l^{6d} = 30$ and by 87% for $l^{6d} = 50$. The final meshes are more anisotropic compared to the initial one. We increase the global aspect ratio q_{Γ_h} from 1.21 to 95.83 and to 65.04, for $l^{6d} = 30$ and $l^{6d} = 50$, respectively.



initial mesh final mesh

Figure 12: The initial bunny mesh (left) and the resulting anisotropic adapted one (right).



initial mesh final mesh

Figure 13: A detail of the initial bunny mesh (left) and the same detail for the anisotropic adapted one (right).

5.4 Real-life Examples

Finally, we study two real-life applications from magnetic resonance imaging processes.

Bronchus

This mesh represents the trachea and the main branches of the bronchi [11]. The geometry is rather complex. It consists of a sequences of branches that become smaller at each bifurcation. Triangles in the initial mesh are not aligned according to the curvature of the geometry, see Figure 16 and the detail in Figure 17.

The surface Γ_0^s is built using thin plate splines, $\varepsilon = 0.1$, $K = 200$, $Ov = 3$.

initial mesh $l^{6d} = 30$ $l^{6d} = 50$

Figure 14: The "Lucy" mesh with 510 180 elements (without edges) as well as the thinned out meshes for $l^{6d} = 30$ and 141 674 elements as well as for $l^{6d} = 50$ and 67 888 elements.

initial mesh $l^{6d} = 30$ $l^{6d} = 50$

Figure 15: A detail of the "Lucy" mesh which is thinned out according to different target edge lengths.

We set σ to 1 and consider three different target edge lengths

$$l^{6d} \in \{0.25,\, 0.5,\, 1.0\}\,.$$

After applying our remeshing procedure the previously problematic triangles disappear and are replaced by new curvature-aligned ones. This example underlines the flexibility of the proposed adaptation procedure. In fact, it is possible to achieve different tasks by tuning the parameter l^{6d}. On the one hand, if we consider a large value for l^{6d}, the remeshing process becomes a mesh simplification method, see the detail in Figure 17 corresponding to $l^{6d} = 1.0$. On the other hand, if we consider a relatively short target edge length, our algorithm creates a surface mesh that is smoother and finer than the initial one, see the detail in Figure 17 corresponding to $l^{6d} = 0.25$. Moreover, if we tune the target length in such a way that we obtain approximately the same amount of elements as in Γ_h^{ini}, the computational effort to deal with this mesh is the same as before but its triangles are curvature-aligned. See Figure 17 with target edge length $l^{6d} = 0.5$.

This observation is numerically verified by the data in Table 3. Here we state the number of elements in the initial and adapted meshes for different target lengths. In this table, we additionally provide the values of the global aspect ratio q_{Γ_h} which quantify the degree of anisotropy.

	initial mesh	$l^{6d} = 0.5$	$l^{6d} = 1.0$	$l^{6d} = 0.25$
#ele.	42 692 100%	34 954 82%	12 026 28%	132 784 311%
q_{Γ_h}	6.00e+01	1.28e+02	7.53e+02	7.08e+02

Table 3: The number of elements for the bronchus meshes. We provide the percentage of the triangles in the mesh with respect to the number of triangles in the adapted meshes as well as the global aspect ratios.

Vertebrae

Finally, we look at a mesh which represents one vertebra of the dorsal spine. The triangles in the Γ_h^{ini} are highly irregular. Many regions are too coarse and nonsmooth, see the initial mesh in Figure 18 and the detail in Figure 19.

The surface Γ_0^s is built using thin plate splines, $\varepsilon = 0.1$, $K = 19$, $Ov = 2$. We set σ to 5 and consider three different edge lengths

$$l^{6d} \in \{0.5,\, 1.5,\, 3.0\}\,.$$

The triangles in the resulting meshes are perfectly oriented and stretched to fit the input geometry, which can be seen in Figure 19. Even though some regions in the input mesh are too coarse and miss some details, in the final adapted meshes these undesirable features completely disappear, see Figure 19. Moreover, the smaller l^{6d}, the smoother the approximation of the input surface.

6 Conclusion

We presented a new anisotropic surface remeshing algorithm, which can be used to improve problematic inputs coming from discrete surface data sets. The

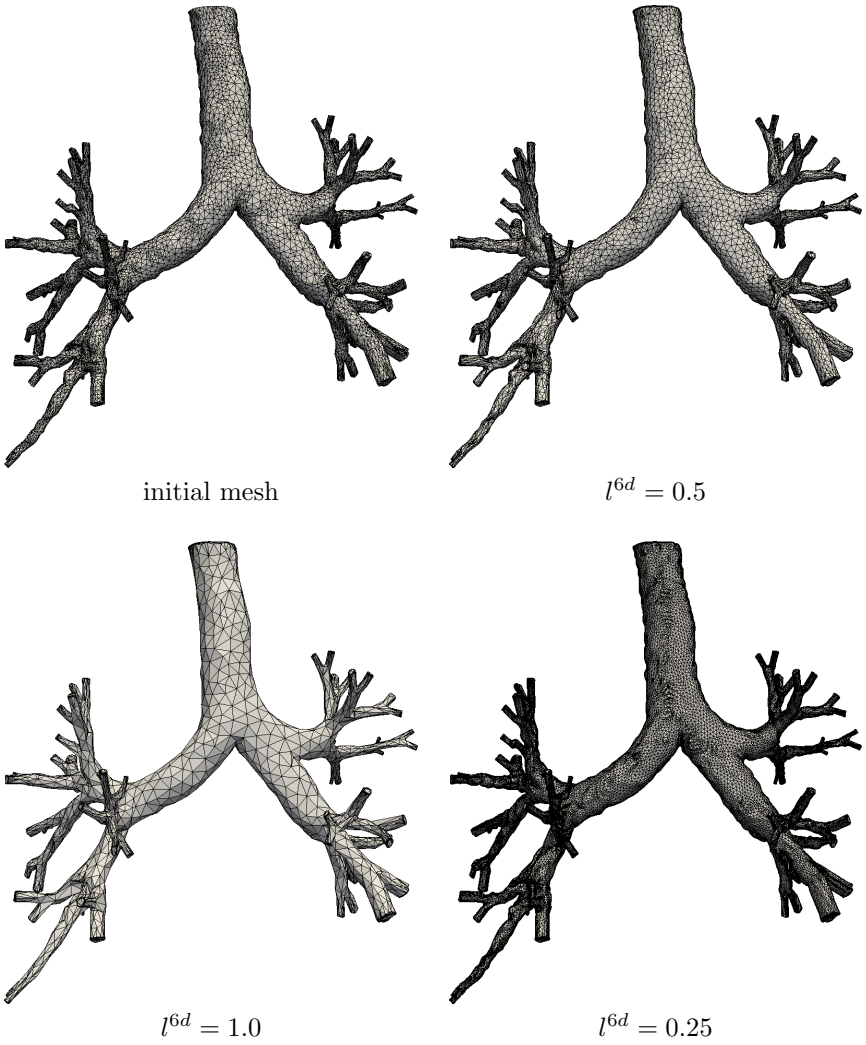


Figure 16: The initial bronchus mesh, courtesy Fetita *et al.* [11], and the adapted ones for different target edge lengths.

anisotropy is guided by the curvature of the surface. Depending on the target edge length parameter, our new method becomes either a mesh simplification, a surface remeshing or a fill-in algorithm. In particular, when the initial mesh is extremely coarse, the algorithm increases the resolution of the poor initial data. The influence of the different parameters in the algorithm were studied and tested on several classical and even real-life meshes. In this paper, we assume that the discrete data represent a smooth surface, future research may look at discrete data which contain sharp features.

Acknowledgments

The work of Franco Dassi was partially supported under the "Leibniz - DAAD Research Fellowship 2014".

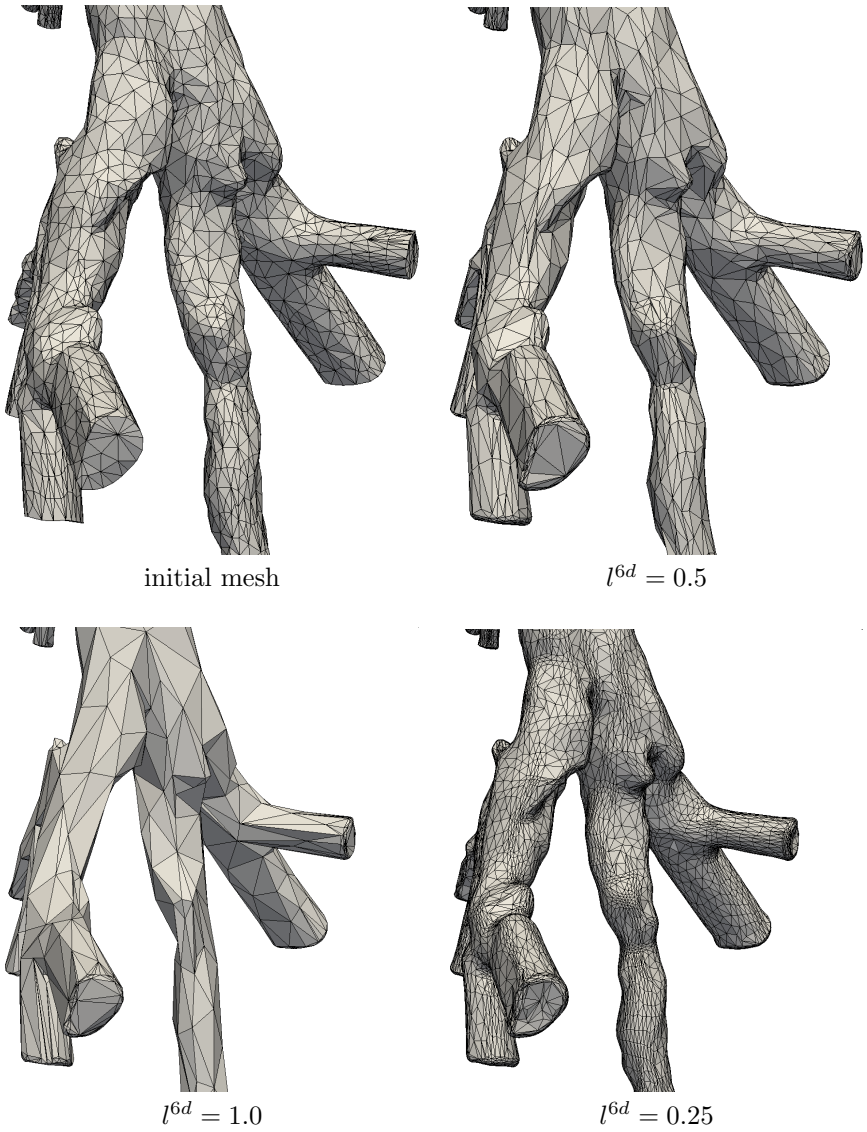


Figure 17: A detail of the initial bronchus mesh and the adapted ones for different target edge lengths.

References

[1] F. J. Bossen and P. S. Heckbert, *A pliant method for anisotropic mesh generation*, in Proceedings of the 5th International Meshing Roundtable, Albuquerque, NM, 1996, Sandia National Laboratories, pp. 63–74.

[2] G. D. Cañas and S. J. Gortler, *Surface remeshing in arbitrary codimensions*, Vis. Comput., 22 (2006), pp. 885–895.

[3] F. Dassi, A. Mola, and H. Si, *Curvature-adapted remeshing of CAD surfaces*, Procedia Engineering, 82 (2014), pp. 253 – 265, doi:`http://dx.doi.`

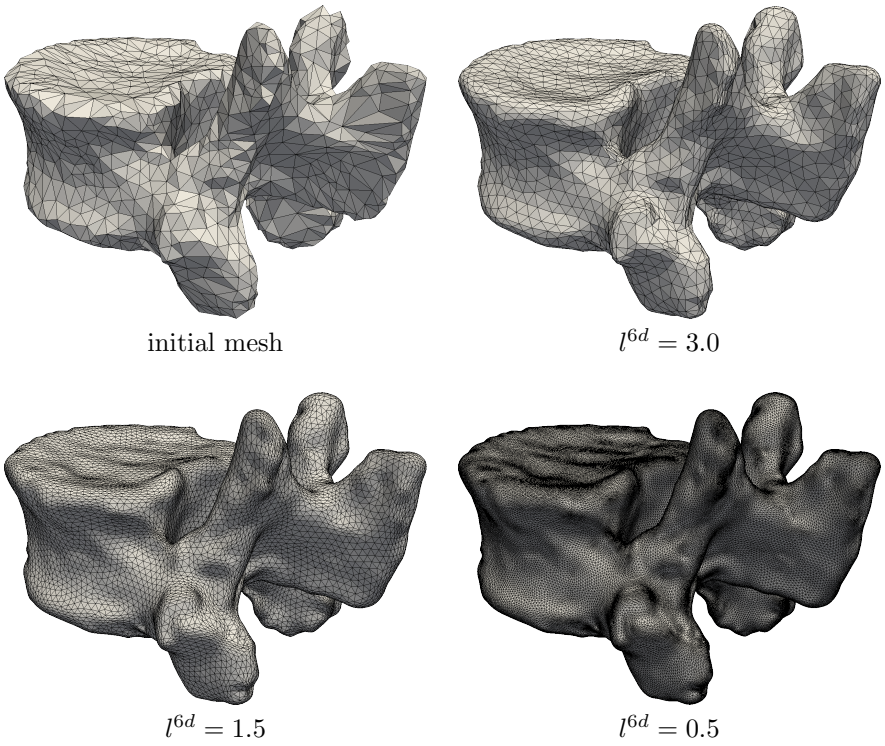


Figure 18: The initial vertebrae mesh and the adapted ones with different target edge lengths.

org/10.1016/j.proeng.2014.10.388, http://www.sciencedirect.com/science/article/pii/S1877705814016671. 23rd International Meshing Roundtable (IMR23).

[4] F. Dassi, S. Perotto, L. Formaggia, and P. Ruffo, *Efficient geometric reconstruction of complex geological structures*, Mathematics and Computers in Simulation, 106 (2014), pp. 163 – 184, doi:http://dx.doi.org/10.1016/j.matcom.2014.01.005, http://www.sciencedirect.com/science/article/pii/S0378475414000184.

[5] F. Dassi and H. Si, *New Challenges in Grid Generation and Adaptivity for Scientific Computing*, Springer International Publishing, Cham, 2015, ch. A Curvature-Adapted Anisotropic Surface Re-meshing Method, pp. 19–41, doi:10.1007/978-3-319-06053-8_2, http://dx.doi.org/10.1007/978-3-319-06053-8_2.

[6] H. L. de Cougny and M. S. Shephard, *Surface meshing using vertex insertion*, in Proceedings of the 5th International Meshing Roundtable, Citeseer, 1996, pp. 243–256.

[7] H. Edelsbrunner, *Geometry and topology for mesh generation*, Cambridge University Press, 2001.

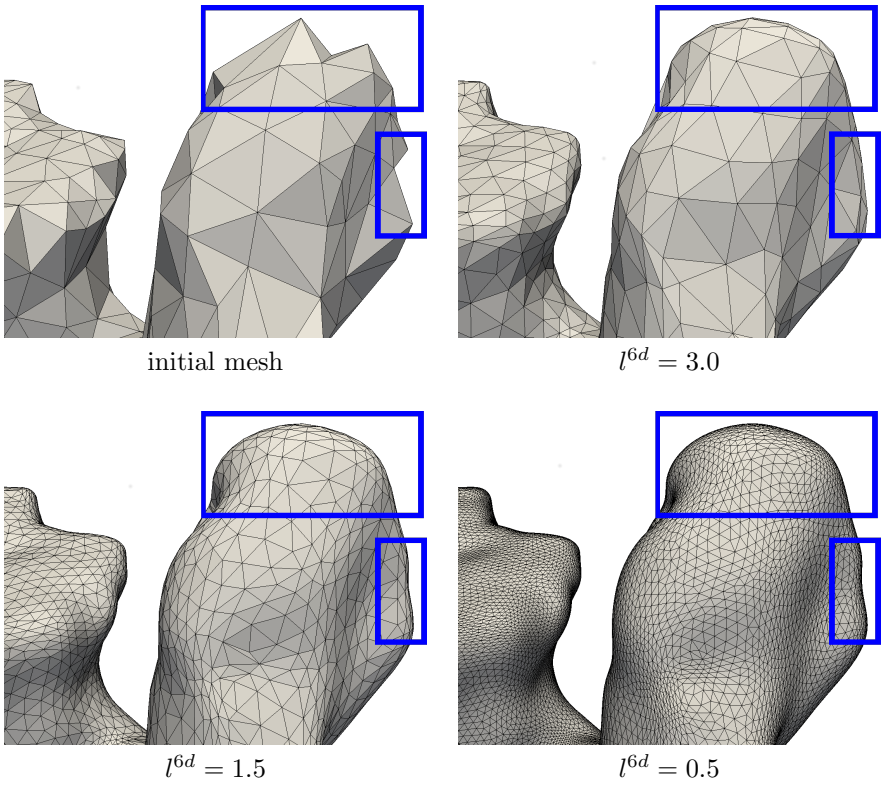


Figure 19: A detail of the initial vertebrae mesh and its corresponding adaptations for different target edge lengths.

[8] J. Escobar, G. Montero, R. Montenegro, and E. Rodríguez, *An algebraic method for smoothing surface triangulations on a local parametric space*, International Journal for Numerical Methods in Engineering, 66 (2006), pp. 740–760.

[9] P. Farrell and H. Wendland, *RBF Multiscale Collocation for Second Order Elliptic Boundary Value Problems*, SIAM J. Numer. Anal., 51 (2013), pp. 2403–2425.

[10] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, vol. 6 of Interdisciplinary Mathematical Sciences, World Scientific Publishing, Hackensack, 2007.

[11] C. Fetita, S. Mancini, D. Perchet, F. Preteux, M. Thirie, and L. Vial, *An image-based computational model of oscillatory flow in the proximal part of tracheobronchial trees*, Comput. Methods Biomech. Biomed. Engin., 8 (2005), pp. 279 – 293.

[12] L. Formaggia, S. Micheletti, and S. Perotto, *Anisotropic mesh adaptation in computational fluid dynamics: Application to the advection-diffusion-reaction and the Stokes problems*, Appl. Numer. Math., 51 (2004), pp. 511–533, doi:10.1016/j.apnum.2004.06.007, http://dx.doi.org/10.1016/j.apnum.2004.06.007.

[13] L. Formaggia and S. Perotto, *New anisotropic a priori error estimates*, Numer. Math., 89 (2001), pp. 641–667.

[14] P. J. Frey, *Generation and adaptation of computational surface meshes from discrete anatomical data*, International Journal for Numerical Methods in Engineering, 60 (2004), pp. 1049–1074.

[15] P. J. Frey and F. Alauzet, *Anisotropic mesh adaptation for CFD computations*, Comput. Meth. Appl. Mech. Engrg., 194 (2005), pp. 5068–5082.

[16] P. J. Frey and H. Borouchaki, *Geometric surface mesh optimization*, Comput. Vis. Sci., 1 (1998), pp. 113–121.

[17] E. Hartmann, *On the curvature of curves and surfaces defined by normalforms*, Computer Aided Geometric Design, 16 (1999), pp. 355 – 376, doi:http://dx.doi.org/10.1016/S0167-8396(99)00003-5, http://www.sciencedirect.com/science/article/pii/S0167839699000035.

[18] H. Hoppe, *Progressive meshes*, in Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ACM, 1996, pp. 99–108.

[19] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, *Mesh optimization*, in Proceedings of the 20th annual conference on Computer graphics and interactive techniques, ACM, 1993, pp. 19–26.

[20] A. Iske, *Multiresolution Methods in Scattered Data Modelling*, vol. 37 of Lecture Notes in Computational Science and Engineering, Springer, Berlin, 2004.

[21] B. Lévy and N. Bonneel, *Variational anisotropic surface meshing with voronoi parallel linear enumeration*, in Proceedings of the 21st International Meshing Roundtable, Springer, 2013, pp. 349–366.

[22] E. Marchandise, C. Piret, and J.-F. Remacle, *Cad and mesh repair with radial basis functions*, Journal of Computational Physics, 231 (2012), pp. 2376 – 2387, doi:http://dx.doi.org/10.1016/j.jcp.2011.11.033, http://www.sciencedirect.com/science/article/pii/S0021999111006899.

[23] J. R. Shewchuk, *What is a good linear finite element? - interpolation, conditioning, anisotropy, and quality measures*, tech. report, In Proc. of the 11th International Meshing Roundtable, 2002.

[24] V. Surazhsky and C. Gotsman, *Explicit surface remeshing*, in Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing, Eurographics Association, 2003, pp. 20–30.

[25] H. Wendland, *Piecewise Polynomial, Positive Definite and Compactly Supported Radial Functions of Minimal Degree*, Adv. Comput. Math., 4 (1995), pp. 389–396, doi:10.1007/BF02123482, http://dx.doi.org/10.1007/BF02123482.

[26] H. Wendland, *Scattered Data Approximation*, vol. 17 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2005.