Curvature based remeshing for phase field based topology optimization

Tim Burg

A thesis presented for the degree of
Master of Science

Supervised by:
Professor Dietmar Hömberg
Hang Si

Technische Universität Berlin, Germany
November 2019

I, Tim Burg confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

This work deals with a novel approach to remesh trianglular meshes that facilitates a surface interpolation by Radial Basis Fuctions (RBFs). While serving a representation of the surface for remeshing this interpolation also allows to obtain a normalvector to the surface which is used in a higher-dimensional embedding sheme to yield a curvature adapted mesh ie. smaller triangle-sizes where the curvature is larger.

A remeshing-program that incorporates the resulting algorithm was implemented and a parameter study for the interpolation conducted.

Consequently the program was used to remesh the isosurfaces obtained from several phase-field topology optimizations.

Acknowledgements

Interdum et malesuada fames ac ante ipsum primis in faucibus. Aliquam congue fermentum ante, semper porta nisl consectetur ut. Duis ornare sit amet dui ac faucibus. Phasellus ullamcorper leo vitae arcu ultricies cursus. Duis tristique lacus eget metus bibendum, at dapibus ante malesuada. In dictum nulla nec porta varius. Fusce et elit eget sapien fringilla maximus in sit amet dui.

Mauris eget blandit nisi, faucibus imperdiet odio. Suspendisse blandit dolor sed tellus venenatis, venenatis fringilla turpis pretium. Donec pharetra arcu vitae euismod tincidunt. Morbi ut turpis volutpat, ultrices felis non, finibus justo. Proin convallis accumsan sem ac vulputate. Sed rhoncus ipsum eu urna placerat, sed rhoncus erat facilisis. Praesent vitae vestibulum dui. Proin interdum tellus ac velit varius, sed finibus turpis placerat.

Table of Contents

Abstract **i**

Acknowledgements **ii**

1 Introduction & Overview **1**

1.1 common approaches for remeshing 2

1.2 Radial basis function surface interpolation 2

1.2.1 Higher dimensional embedding for curvature adaption 2

2 Theoretical backgrounds **3**

2.1 Topology optimization via the phase-field method 3

2.1.1 The equations of static elasticity 3

2.1.2 Stess, strain and the equations of equilibrium in the linear case 5

2.1.3 Variational formulation 6

2.1.4 Compliance minimization 7

2.1.5 Phase-field formulation and advancing-front algorithms 7

2.1.6 Isosurface extraction 7

2.2 Radial-basis-function theory 8

2.2.1 RBF Interpolation 8

2.2.2 Existence and Uniqueness results 9

2.2.3 Commonly used Radial basis functions 10

2.2.4 Scaling of RBF functions, ambiguities and interpolation properties 11

2.2.5 surface interpolation 12

2.3 Remeshing operations 13

2.3.1 Edge collapse 14

2.3.2 Edge split 15

2.3.3 Edge flip 16
2.3.4 Vertex smoothing 17
2.3.5 Projection of vertices onto the surface 18
2.4 Higher dimensional embedding 18

3 Algorithm description and implemenation details 20

4 Results 21
4.1 Parameter survey 22
4.2 convergence behaviour 22

5 Problems, outlook and future work 23

6 References 24

Chapter 1

Introduction & Overview

Triangular meshes are the most prominent representation of 3D surfaces in computer graphics and the go-to format for computer-aided-design. Aside from the topological aspects of the described surfaces or non-manifold errors that can cause issues with remeshing and additive manufacturing, These meshes, being piecewise linear, can only approximate curved surfaces and often such approximations may be severely ill-conditioned for computations and inefficient in their surface-description. This for example the case when the mesh is generated from scattered data. If this is the case then remeshing techniques can be used to yield a better mesh.

One such case which is the one treated here arises after the extraction of an isosurface-mesh of a scalar-valued function defined on a finite-element mesh. The isovalue-intersections may cut the Finite-element-simplices close to corners and edges resulting in very small and possibly distorted ie. non-equilateral triangles.

The Finite-element application considered in this work is a topology optimization with the phase-field method. The phase-field distinguishes material from void but has a continuous range from 0 to 1 where 0 is void and 1 is material. Usually then the isosurface to the value 0.5 is used to determine the boundary of the solid-body.

1.1 common approaches for remeshing

There are different approaches for remeshing a given triangular mesh that each involve some form of interpolation or approximation to find new vertices for the mesh or generate a whole new mesh altogether. Interpolation schemes do, as the name implies respect the original data(here: the mesh) in the sense that the vertices are still points on the surface.

1.2 Radial basis function surface interpolation

For remeshing a surface one has to have a representation that can be queried for new on-surface points. Here implicit surface descriptions, ie. $F(x) = 0$ for an interpolating space-function $F : D \rightarrow \mathbb{R}$ where D is the data space have proven to be practical. Radial-basis-functions interpolation falls into that catergory and it is, besides polynomial-interpolation, among the most widely used interpolation-scheme nowadays. Radial-basis-functions are especially suited for multivariate data (read: arbitary dimensions) as they only depend on the space-dimensions through the associated vectornorm. The interpolant is given as a simple sum over basis funtions centered at the data points i:

$$F(x) = \sum_i \alpha_i \varphi(\|x - x_i\|)$$

1.2.1 Higher dimensional embedding for curvature adaption

The RBF-interpolant can be derived analytically to obtain the gradient. This gradient, standing perpendicular on the 0-level-set describes the curvature of the mesh and used for curvature adaptive remeshing. This is accomplished with something called a higher dimensional embedding. The idea is that

Chapter 2

Theoretical backgrounds

2.1 Topology optimization via the phase-field method

2.1.1 The equations of static elasticity

Mathematical elasticity can be considered a branch of continuum dynamics whose research reaches as far back as the late 16th century. I only give a short outline of the stepstones to linear elasticity and refer mostly to the book mathematical elasticity by Ciarlet which is a comprehensive standard piece on the topic.

Continuum dynamics deals with a body occupying a lipschitz-continuous reference configuration $\overline{\Omega} \subset \mathbb{R}^3$ under rest which is deformed to a configuration $\Omega \subset \mathbb{R}^3$ by applied forces. The deformation is then described by an injective mapping φ which contains a displacement field $u : \overline{\Omega} \mapsto \Omega$:

$$\varphi : \overline{\Omega} \mapsto \Omega \qquad \varphi = id + u$$

The deformation and displacement mappings are required to be two times continuously differentiable but this reqirement can be relaxed in the variational formulation of the equations. I denote the coordinates in the reference configuration with x and and those in the deformed configuration with $x^\varphi = \varphi(x)$. In engineering Textbooks those coordinates are sometimes referred to as Lagrange- and Euler-coordinates respectively.

The elasticity theory is then build on the follwowing two contibutions from Cauchy of which the second is fundamental to continuum dynamics:

1. Axiom of force balance:

   Given volume- and surface-force-densities as f^φ and g^φ in the deformed configuration then for every subset $A^\varphi \subset \Omega$ the following equality holds:
   $$\int_{A^\varphi} f^\varphi(x^\varphi)dx^\varphi + \int_{\partial A^\varphi} t^\varphi(x^\varphi, n^\varphi)da^\varphi = 0$$
   Here, dx^φ and da^φ are the volume and surface elements in the deformed configuration, n^φ is the surface-unit-normal and t^φ is the cauchy stress vectorfield:

   $$t^\varphi : \Omega \times \mathbb{S}_1 \mapsto \mathbb{R}^3 \quad where \quad \mathbb{S}_1 := \{v \in \mathbb{R}^3 \mid \|v\| = 1\}$$

   Note cauchy stress vector t^φ depends on the given Volume A only through the normal vector at a surface point and that any surface-force dictated on part of $\partial A \cap \partial\Omega$ must be dispersed through the remaining part of ∂A.

2. Stress Tensor theorem:

   Assuming that f^φ is continuous and $t^\varphi \in \mathrm{C}^1(\Omega) \cap \mathrm{C}(\mathbb{S}_1)$, then t^φ is linear w.r.t. to the second argument ie.:

   $$t^\varphi(x^\varphi, n) = T^\varphi(x^\varphi)n \qquad \forall x^\varphi \in \Omega, \qquad \forall n \in \mathbb{S}_1 \tag{2.1a}$$

   and

   $$-\mathrm{div}^\varphi T^\varphi(x^\varphi) = f^\varphi \qquad \forall x^\varphi \in \Omega \tag{2.1b}$$
   $$T^\varphi(x^\varphi) = T^\varphi(x^\varphi)^T \qquad \forall x^\varphi \in \Omega \tag{2.1c}$$
   $$T^\varphi(x^\varphi)n^\varphi = g^\varphi(x^\varphi) \qquad \forall x^\varphi \in \Gamma^\varphi \tag{2.1d}$$

   where Γ^φ is the part of $\partial\Omega$ where the boundary condition g is prescribed. (See Ciarlet 1990, pp.63–65 for the proof)

Notice that the forumulation above uses the stress tensor in the deformed configuration. The pullback of the tensor onto the reference configuration is achieved with the piola-transform after which it needs to be symmetrized again. This then yields the so-called first and second Piola-Kirchhoff-Stress-Tensors denoted with Σ. The densities in the pullback of the forces is often ignored. These are then called dead loads (see Ciarlet 1990, chap.2.7).

They are omitted here for brevity but the second Piola-Kirchhoff-Stress is the stress tensor to be determined in the next chapter.

2.1.2 Stess, strain and the equations of equilibrium in the linear case

So far the theory is valid for all continuums but there are also nine unknown functions, namely the three components of the deformation and the six components of the stress tensor tensor. Luckily, several simplifictaions can be made in case of isotropic and homogeneous media that lead to a remarkably simple form of the tensor.

To this end the chauchy strain tensor C and its difference from unity E is introduced. They describes the first order local change in length-scale under a deformation and are via the fréchet derivative of the mapping φ, $\nabla\varphi$:

$$\nabla\varphi = \begin{pmatrix} \partial_1 u_1 & \partial_2 u_1 & \partial_3 u_1 \\ \partial_1 u_2 & \partial_2 u_2 & \partial_3 u_2 \\ \partial_1 u_3 & \partial_2 u_3 & \partial_3 u_3 \end{pmatrix}$$

$$C = \nabla\varphi^T\nabla\varphi = I + \nabla u^T + \nabla u + \nabla u^T\nabla u = I + 2E$$

Viewed in a different light, the deformed state can be considered a manifold with C as the metric-tensor.

The simplification of the second Piola-Kichhoff-Stress-tensor follows these steps:

1. The stress tensor can only depend on φ through its derivative $\nabla\varphi$

(Elasticity)

2. Material-Frame Indifference
3. Isotropy
4. Rivlin-Ericksen representation theorem
5. Homogeneity

Details on these steps can again be found in (Ciarlet 1990, chap.3). After following these steps, Σ takes on the following form:

$$\Sigma(C) = \lambda(\mathrm{tr}E)I + 2\mu E + o(\|E\|)$$

Here, λ and μ are the lamé coefficients of the material. In the linear theory that is used as the basis for the topology optimization, the strain E is replaced with the linearized version ε:

$$\varepsilon = \frac{1}{2}\nabla u^T + \nabla u$$

which yields the following even simpler form of the tensor which is referred to as σ:

$$\sigma = \lambda(\nabla u)I + \mu\left(\nabla u + \nabla u^T\right)$$

This is called Hooks-law and usually written with an additional tensor c: Where the last form is written in vector form for the components of the tensors.

2.1.3 Variational formulation

For finite-element simulations and reduced smoothness requirements of the displacement, a variatonal formulation of the equilibrium equations 2.1 must be formulated.

Multiplying equation 2.1b with a test function θ on both sidesa and intgrating yields:

$$\int_{\Omega^\varphi} \mathrm{div}^\varphi T^\varphi \cdot \theta^\varphi dx^\varphi = -\int_{\Omega^\varphi} f^\varphi \theta^\varphi dx^\varphi + \int_{\Gamma^\varphi} g^\varphi \theta^\varphi$$

Using the Greens-fomula for Tensor fields:

$$\int_{\bar{\Omega}} \mathrm{div} H \cdot \theta dx = -\int_{\bar{\Omega}} H : \nabla\theta dx + \int_{\Gamma} Hn \cdot \theta da$$

and applying the pullback to the reference configuration with the second Piola-Kirchhoff-Stress-tensor then gives:

$$\int_{\bar{\Omega}} \nabla\varphi : \Sigma\nabla\theta dx = \int_{\bar{\Omega}} f \cdot \theta dx + \int_{\Gamma} g \cdot \theta da$$

For All sufficiently regular vector fields $\theta : \bar{\Omega} \mapsto \mathbb{R}^3$. This is also called the 'principle of virtual work' (in the reference configuration).

2.1.4 Compliance minimization

2.1.5 Phase-field formulation and advancing-front algorithms

2.1.6 Isosurface extraction

Since the Finite-Element-Mesh is providing a 3D-tesselation of the domain, which in this case consists of tetrahedra, the generation of an isosurface is handled as in the marching-tetrahedra algorithm. Tetrahedra, as opposed to cubes, can only have 3 distinct cases of edge intersections that differ in terms of their makeup of triangular faces. No intersections, intersection at 3 edges(1 triangle) and intersection at 4 edges (2 triangles). See figure 2.1 for an illustration.

For the edge intersections, a linear interpolation of the values between two vertices is used. The intersections are then found via simple line intersections comparable to the section of the x-axis for a line. If 3 intersections are found one triangle is generated with the vertices of the intersections and if 4 intersections are found 2 triangles are generated. Susequently the ordering of the vertices is checked so that looking from the outside, the vertices are ordered counterclockwise in accordance with the stl-specification. For this, the function values at the tetrahedra-nodes are considered to find a point that is inside (has a value greater than 0) This is especially important since the orientation is used in the remeshing procedure and can only be correctly determined at this step.

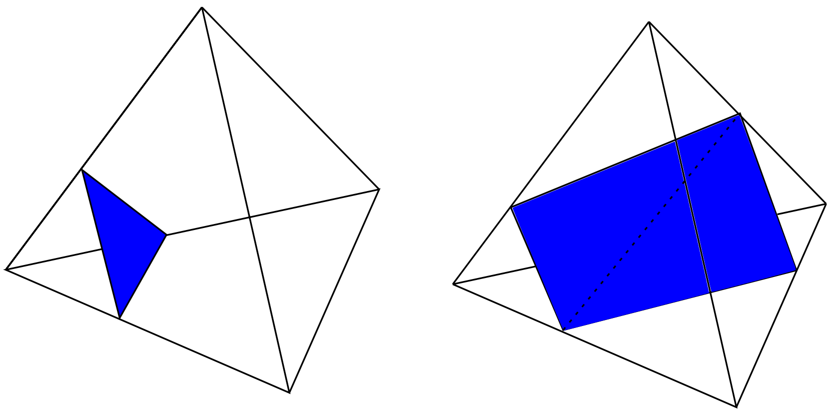

Figure 2.1: In the case of a 3D-Tetrahedra tesselation only 3 distinct cases can appear a)intersection at three edges(left) or b)intersection at 4 edges(right) or c)no intersections(not displayed)

2.2 Radial-basis-function theory

2.2.1 RBF Interpolation

Interpolation can be viewed as a special kind of approximation in which, for an approximant S to some function F, it is demanded that the interpolant reproduces the original functions values at special points x_i ie.:

$$S(x_i) = F(x_i) \quad \forall i \in \Xi \tag{2.2}$$

Where Ξ is some finite (possibly scattered) dataset in $\mathbb{R}^N$ (multivariate) ie. a set $(\xi, f_\xi) \in \mathbb{R}^N \times \mathbb{R}$. The functions considered here are scalar valued. A construction of vector-valued interpolants from scalar-valued conmponentones ist straightforward.

Interpolants are usually constructed from some function space which in this case is made up of Radial-basis-functions. Radial-basis-functions are special in that they allow easy interpolation of scattered multivariate data with guaranteed existence and uniqueness results. In general, different interpolants do behave differently for the space in between the datasites and are distinguished by their approximation and or convergence properties for special classes or cases of F. However, when no original function(just the values $F(x_i)$) is given the accuracy of an interpolation can not generally be assessed. Because this is the case here, determining qualities for the RBF-interpolant are discussed in section ...

Radial-basis-function interpolation constructs the interpolant S as a linear combination of scaled Radial-basis-functions centered at the datasites:

$$S(x) = \sum_i \alpha_i \varphi(\|x - x_i\|)$$

The norm denotes the standard euclidian norm which is essential for the convergence results (see Buhmann n.d. p)
The Radial-basis-functions themselves are functions of the form $\varphi : \mathbb{R} \mapsto \mathbb{R}$

By introducing the interpolation matrix A as:

$$A = \varphi(\|x_i - x_j\|)|_{i,j}$$

we can write the interpolation condition 2.2 as:

$$A\alpha = F$$

2.2.2 Existence and Uniqueness results

The invertbility of the interpolation matrix has been investigated thoroughly in the 1970's and 1980's. Key results rely on complete monotonicity of the Radial-Basis-function, ie. the property:

$$(-1)^l g^l(t) \geq 0 \quad \forall l \in \mathbb{N} \quad \forall t > 0$$

Then it can be shown that the interpolation matrix is always positive definite. The proof relies on the Bernstein representation theorem for monotone functions. (see Buhmann n.d., pp.11–14 for the case of multiquadratics) A weaker requirement is that only one of the derivatives must be completely monotone. This then leads to the concept of conditonally positive definiteness of a function in which a polynomial is added to the interpolant.

- complete monotonicity yields a positive definite A (result due to Michelli 1986)
- weaker concept: complete monotonicity of some derivative: $(-1)^k \frac{d^k}{dt^k}\varphi(\sqrt{t})$
  - introduces conditionally positive definite functions that use an added polynomial that vanishes on the data sites for interpolation and thus the interpolant is again unique and exists
- called "unisolvence"

2.2.3 Commonly used Radial basis functions

It remains for us to recite some of the more often used RBFs and state if they are positive definite or conditionally so. During the course of writing this thesis it became clear that only local basis functions would be good candidates for a feasible surface interpolation due to the number of vertices common in triangular meshes.

Now commonly used are the Wendland functions (see Wendland 1995) which are piecewise polynomial, of minimal degree and positive definite. For the surface interpolation I use the C2 continuous function and it's derivative.

Table 2.1: RBF functions with global support

function	name	definiteness
e^{-r^2}	gaussian	pd
$\sqrt{r^2+1}$	multiquadratics	pd
$1/\sqrt{r^2+1}$	inverse multiquadratics	pd
r^3	polyharmonic spline	cpd

Table 2.2: Local RBF functions introduced by Wendland (Wendland 1995)

function	name	definiteness
$(1-r)_+^4(4r+1)$	$\varphi_{3,1}(r)$	pd
$(1-r)_+^6(35r^2+18r+3)$	$\varphi_{3,2}(r)$	pd
$(1-r)_+^8(32r^3+25r^2+8r+1)$	$\varphi_{3,3}(r)$	pd

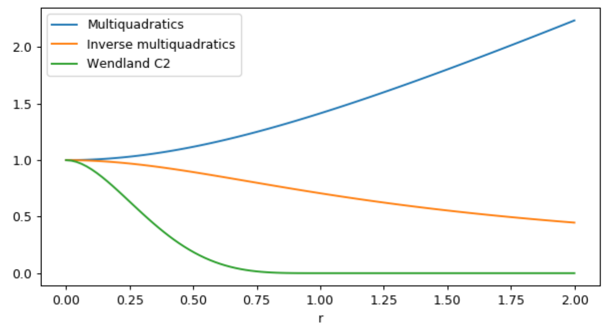

Figure 2.2: Comparison of different RBF functions. Note that a convergence to zero is not mandatory. However, the Wendland functions become zero after r=1

2.2.4 Scaling of RBF functions, ambiguities and interpolation properties

The normal RBF functions have a fixed spread as seen in fig. 2.2. Since spacing of the interpolation data is not fixed, a scale needs to be introduced that scales r such that the RBFs extend into the space between the datasites. Otherwise the interpolant might just have, in the exteme case, spikes at the sites to attain the required values. To this end I scale r by $r' = r/c$ with a scale parameter c since that makes the Wendland functions extend to exactly the value of this parameter.

This scaling parameter, in general can be nonuniform over the interpolated values but this comes with uncertainty for the solvability of the interpolation system.

Moreover, it cannot be generally stated which value of a scale parameter is more accurate in an interpolation unless there is a target to which the interpolant can be compared. See fig. 2.4

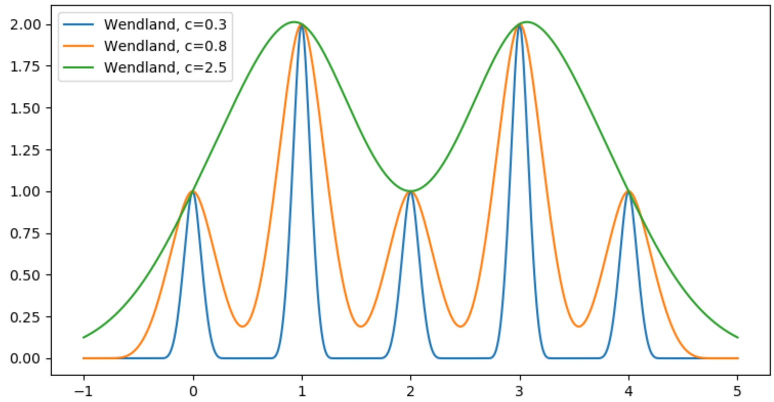

Figure 2.3: Wendland C2 functions for different scaling parameters c. The interpolation values were set to (1,2,1,2,1) at (0,1,2,3,4).

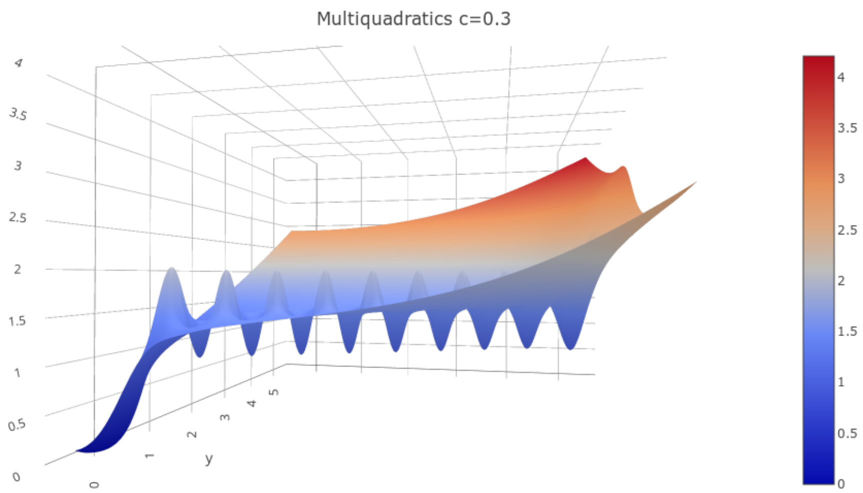

Figure 2.4: Different Radial-Basis-Funtions have different behaviours for off-site values. Multiquadratics grow toward infinity. Displayed is a 1-2 comb in two dimensions

2.2.5 SURFACE INTERPOLATION

Surface descriptions are either explicit or implicit. Explicit means that the surface is the graph of a function $F : \Omega \subset \mathbb{R}^2 \mapsto \mathbb{R}^3$ which can be very complicated to construct. Especially complicated topologies this can usually be only done via 2d-parametric patches of the surface which have their own difficulties for remeshing. Implicit surfaces on the other hand are defined via a functions level set (usually the zero level) ie. $F(x) = 0$ which is easier to construct but is harder to visualise. Usually then for visualization either marching-cubes or raytracing methods are used.

For the surface interpolation with an implicit function this translates to the interpolant being zero at the datasites: $S(x_i) = 0$. Since the zero function would be a trivial solution to this, off-surface constraints must be given. This is usually done with points generated from normalvectors to the surface that are given the value of the signed distance function ie. the value of the distance to the surface:

$$S(\mathbf{x}_i + \epsilon\mathbf{n}_i) = F(\mathbf{x}_i + \epsilon\mathbf{n}_i) = \epsilon \tag{2.3}$$

If not available, the normalvectors can be generated from a cotangent plane that is constructed via a principal component analysis of nearest neighbors. This however is a nontrivial problem. In my case the vectors could be obtained from an average of the normals of the adjacent triangles scaled with the inverse of the corresponding edgelengths:

$$\mathbf{n} = \sum_{T \in \mathcal{N}_T} \frac{1}{\|\mathbf{n}_T\|}\mathbf{n}_T$$

These offset-points were generated for every vertex of the original mesh and in both directions (on the inside and on the outside) such as to give the interpolant a constant slope of one around the surface. This is done to have an area of convergence for a simple gradient-descent projection algorithm.

2.3 Remeshing operations

Different approaches exist to remesh a surface. Most fall into one of the following categories: - triangulate a commpletely new mesh, usually with delauney triangulation and go from there - incremental triangulation, with new nodes inserted or removed one at a time. - local mesh modifications /

pliant remeshing

Additionally most methods utilize some form of vertex-smoothing as this is an straightforward iterative procedure that improves themesh globally and is guaranteed to converge.

The approach used here falls into the latter category and uses consecutive loops of local mesh modifications of the following kinds: - Edge collapse - Edge split - Edge flip - Vertex smoothing

Which of the modification is applied depends on an edges length in comparison to a target-edge-length.

2.3.1 Edge collapse

Edge collapse, as the name suggests removes an edge from the mesh thereby deleting two adjacent triangles and removing one point. Special conditions have to be checked as there are certain configurations that would result in an illegal triangulation. See figures fig. 2.6 and 2.7 To avoid having to project a new midpoint to the surface, the two vertices of the edge are joined at either one of them.

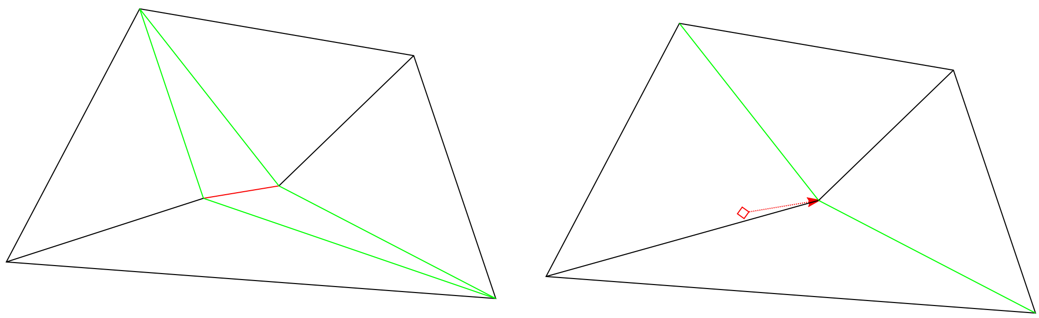

Figure 2.5: Edge collapse with the new point at one of the endpoints

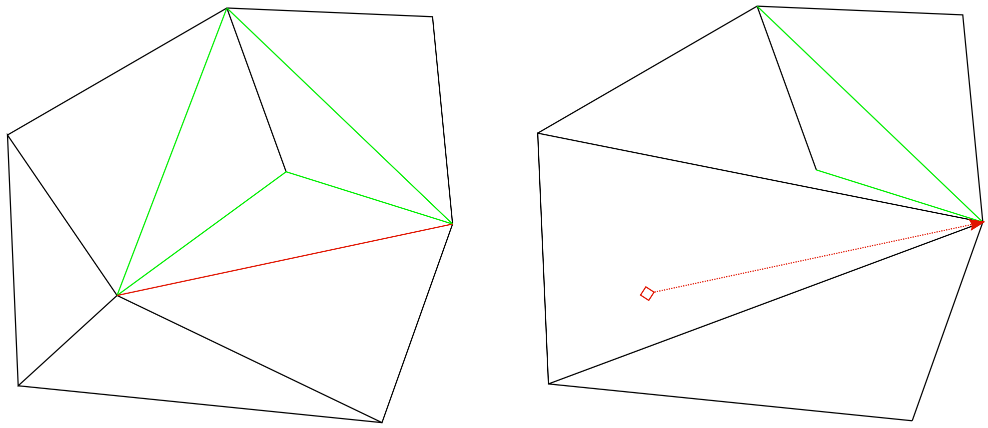

Figure 2.6: Illegal edge collapse with more than two common neighbors for the edges endpoints

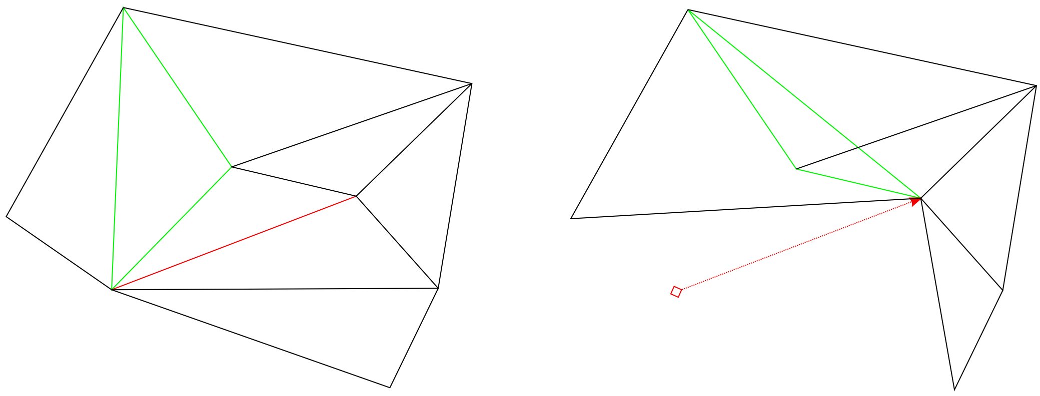

Figure 2.7: Illegal edge collapse with a triangle flip

2.3.2 Edge split

The edge split is a straightforward operation as no special cases have to be taken care of. A new vertex is put at the surface projected midpoint of the existing edge and 4 new edges as well as 4 new triangles replace the split edge and it's adjacent triangles.

2.3.3 Edge flip

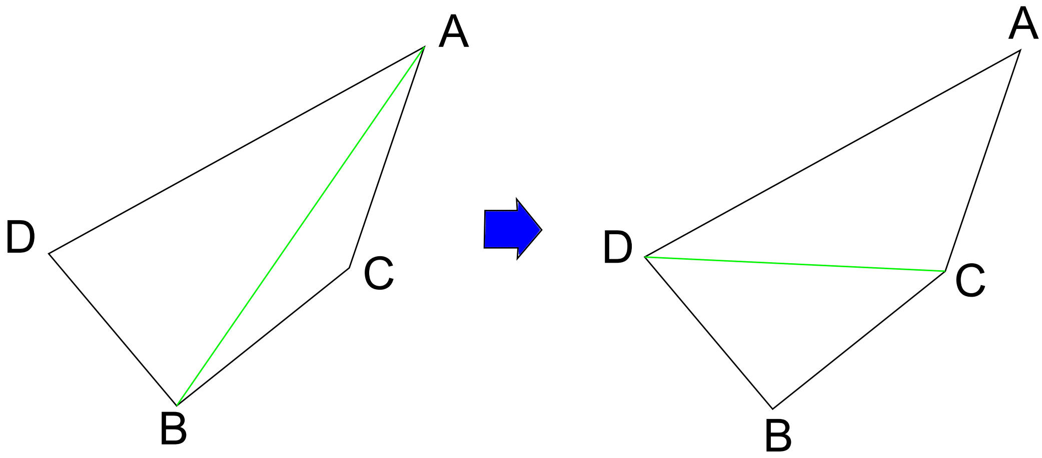

Figure 2.8: Edge flip

An edge flip can dramatically increase the aspect ratio of a triangle if the right conditions are met. Consider the edge in figure **??** Such an edge is flippable if:

- The edge does not belong to the boundary of the mesh
- The edge CD does not already belong to the mesh
- $\phi_{ABC} + \phi_{ABD} < \pi \quad \text{and} \quad \phi_{BAC} + \phi_{BAD} < \pi$
- The angle between the normals of the triangles is not too big to not cast "ridges"

I do a flip based on the following criteria:

2.3.4 Vertex smoothing

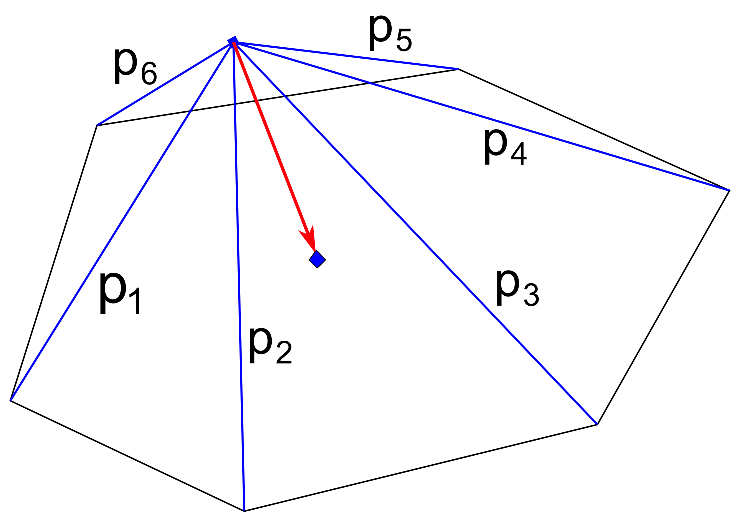

Figure 2.9: Vertex smoothing

Vertex smoothing finds a new position for a given vertex based on the distance to its neighbors according to the following formula:

$$\mathbf{p}' = \mathbf{p} + \alpha \sum_{j \in \mathcal{N}} f(\|\mathbf{p} - \mathbf{p}_j\|)(\mathbf{p} - \mathbf{p}_j)$$

Wherein $\mathcal{N}$ stands for the neighbors, α is a normalization constant and f is a weight function. Different weights have been investigated in (Bossen & Heckbert n.d.) where they constructed a well performing weight function. Given a target edge length t and an actual edge length l a normalized edge length is defined as $d = l/t$ and the weight function reads:

$$f(d) = (1 - d^4) \cdot e^{-d^4}$$

This function pushes if $l < t$ and slightly pulls if $t > l$. The function is plotted in figure 2.10 versus the frequently used laplace weights. Additionally, I clipped the movedistance to 80% of the minium of the adjacent triangles heights. This is done because moves that exceed this distance are likely to cause unacceptable triangles. What unacceptable means is defined in the algorithm section but is basically implemented as triangles with excess tilt versus the surface normal.

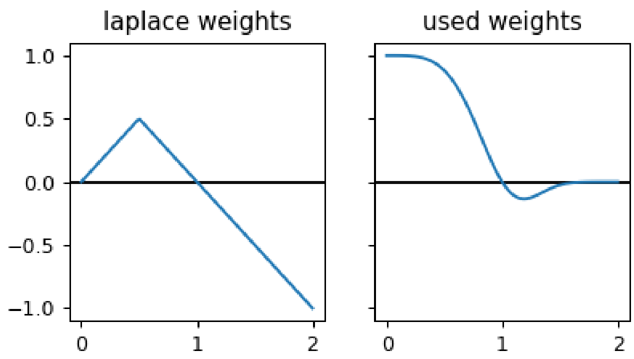

Figure 2.10: The weight function used compared to the laplace weights

2.3.5 Projection of vertices onto the surface

Both in an edge split as well as in vertex smoothing a constructed new vertex must be projected onto the surface. To this end I use a simple gradient descent with a fixed steplength of one. This is a rather heuristical result that has proven much better convergence than the exact steplength. This is most probably due to the fact that around the surface the slope of the function is one by construction.

2.4 Higher dimensional embedding

The term higher dimensional embedding may sound a bit exaggerated for what is actually done. Namely, the pointnormals are included in an edges length calculation as to enlarge the edge when the normals differ. Thereby, the the enlarged edges are remeshed more finely. Formally this reads as follows. Given a vertex x on the surface, it is concatenated with the surface normal n at this point:

$$\Psi(x) = (x, y, z, \sigma n_x, \sigma n_y, \sigma n_z)^T$$

Here σ is a parameter of the embedding and in effect controls how much an edge will be enlarged. With this new Ψ the edgelength between two points a and b will now be defined as:

$$l_{ab}^{6d} = \|\Psi(a) - \Psi(b)\| = \sqrt{(\Psi(a) - \Psi(b), \Psi(a) - \Psi(b))}$$

And in the same manner an angle between the points a, b, c is defined via:

$$cos(\theta_{abc}^{6d}) = \frac{(\Psi(a) - \Psi(c), \Psi(b) - \Psi(c))_{6d}}{l_{ac}^{6d} l_{bc}^{6d}}$$

Chapter 3

Algorithm description and implemenation details

Chapter 4

Results

As mentioned before there is a fundamental challenge in the surface interpolation with radial basis functions that is due to the fact that the surface is given implicitly by the zero-level of the 3-dimensional interpolant. Since the interpolant is only guaranteed to have a zero crossing at the interpolation points and not in between them, the surface can be non-contiguous. More precisely there is no topological guarantee for a manifold surface over a cluster of islands.

Sevel parameters influcence that situation. Those being:

- the spacing and values of the offset points and if they are uniform or not
- the scale-factor(s) of the radial-basis-functions and if they are uniform or not

To assess the acceptance of different parameter-combinations in that regard I conducted a parameter-survey. The aim was to have a general heuristic for an always working or at least 'as good as it gets' parameter set for the following remeshings.

As a essential feature the values of the RBF-interpolant along an outward line through the triangle-centroids were probed for: a) the existence of a zero-crossing (mandatory) b) the witdh between minima and maxima adjacent to the zero crossing (convergence area of the projection)

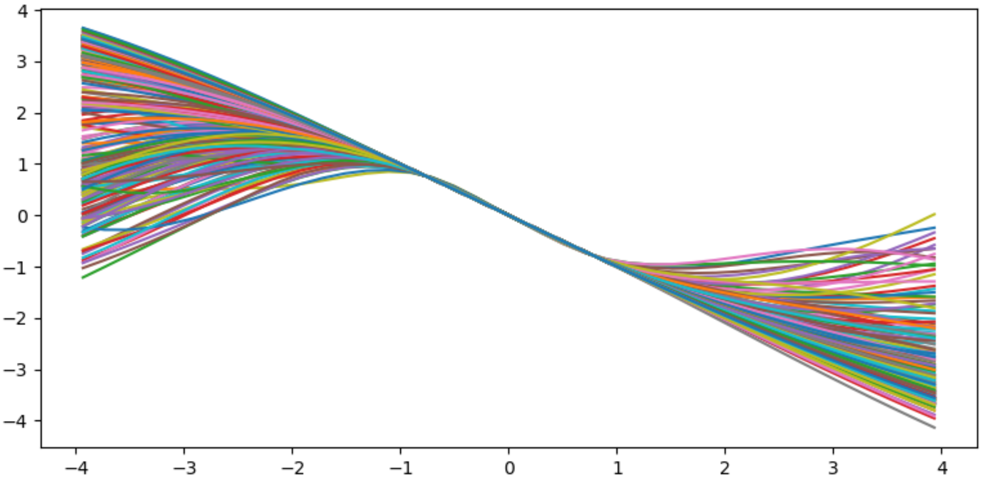

Figure 4.1: At the vertices of the mesh in between the offset interpolation points the RBF-interpolant is well behaved. The offset interpolation points are located at $\pm$ 0.78 in units of the actual mesh

4.1 Parameter survey

4.2 convergence behaviour

Chapter 5

Problems, outlook and future work

Combine interpolation and smoothing into one step with a lower order RBF function that is fitted rather than interpolates. See Buhmann Chapter 8 or >Reconstruction and Representation of 3D Objects with Radial Basis Functions paper<

Chapter 6

References

Bossen, F.J. & Heckbert, P.S., A pliant method for anisotropic mesh generation., p.12.

Buhmann, M.D., Radial basis functions: Theory and implementations., p.271.

Ciarlet, P.G., 1990. Mathematical elasticity, volume i: Three-dimensional elasticity. *Acta Applicandae Mathematica*, 18(2), pp.190–195. Available at: https://doi.org/10.1007/BF00046568 [Accessed September 29, 2019].

Wendland, H., 1995. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4, pp.389–396.