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

In this work I implement and extend a triangular mesh remeshing approach by (Dassi et al. 2016) 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.

Some fundamental issued with the approach are presented and discussed.

As an application, I wrote a remeshing-program that incorporates the algorithm with some corrections to remesh isosurfaces that I obtained from phase-field topology optimization calculations.

The topology optimizations were calculated with a software library pdelib custom to the Weierstrass-Institute.

Both the calculation and the remeshing were conducted on the Weierstraß-servers.

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

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 (which are an issue of their own in remeshing and additive manufacturing), these meshes, being piecewise linear, can only approximate curved surfaces. Often such approximations are inefficient in terms of their required data storage because the grid is isotropic and uses about the same number of datapoints in areas of high curvature variation as in flat areas. This problem arises for example when the datapoints are obtained from some form of measurement data (e.g.laser scanners) or are constructed from local algorithms like marching cubes or marching tetrahedra.

One such case which is the one treated here concerns 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. Variance of the surface is not accounted for in the usual isosurface algorithms and hence a mesh that is as dense as the FE-mesh is extracted. Subsequently various remeshing techniques may be used to yield an adaptive mesh. The method used for this in this work is a pliant remeshing algorithm with local mesh modification.

The application of which the FE-mesh stems from is a topology optimiza-

tion using the phase-field method. Topology optimization is a form of optimization of mechanical structures that can nucleate and adapt holes in the material where low stress is encountered and add material in high stress regions.

Often a certain minimum stiffness or maximum give in a structure is required for it to perform its target application. Topology optimization can then yield designs that need less material or give better stiffness and hence a better performing part for the same material.

The phase-field is integral in this method and 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. Oftentimes the original piecewise-linear complex ie. the original mesh is used to find the new vertices which is linear interpolation. For applications where a smooth output mesh is desired this is not suitable and more intricate interpolation schemes need to be used. However, all Interpolation schemes do respect the original mesh in the sense that the vertices are still points on the surface. This is not the case if the original mesh is approximated ie. fitted with a functional description.

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 $F : D \to \mathbb{R}$ 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 because the dimensionality of the data is only incorporated through the vectornorm of the associated space and as such is especially easy to implement.

To see this, consider the simplest case where the interpolant is given as a 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 easily differentiated analytically to obtain the gradient formula. This gradient, standing perpendicular on the 0-level-set describes the curvature of the mesh and is used for curvature adaptive remeshing. This is accomplished with an extension that incorporates the gradient in a 6d formulation: $(x, y, z, \sigma n_x, \sigma n_y, \sigma n_z)$ The 6d norm is then used in the remeshing process. This essentially gives longer edges where normals are different and induces a refinement.

Final thoughts on the reliability of this method can be found in the conclusive statement.

Chapter 2

Theoretical backgrounds

First I present the theory needed for the topology optimization used. This includes a brief recapitulation of linear elasticity, compliance minimization and then some elaborations on the topology optimization and it's optimality conditions and the numerics that follow from it. After that the extration of the isosurface is described and the basic remeshing operations are introduced.

2.1 Linear elasticity

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 described by an injective

mapping φ via a displacement field $u : \overline{\Omega} \mapsto \Omega$:

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

For the static case treated here, the deformation is time independent. 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 that the 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 surface normal 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 additionally

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

$$T^\varphi(x^\varphi) = T^\varphi(x^\varphi)^T \qquad \forall x^\varphi \in \Omega \tag{2.1c}$$

$$T^\varphi(x^\varphi)n^\varphi = g^\varphi(x^\varphi) \qquad \forall x^\varphi \in \Gamma_g^\varphi \tag{2.1d}$$

where Γ_g^φ is the part of $\partial\Omega$ where the boundary condition g is prescribed and $\mathrm{div}T = \partial_j T_{ij}, e_i$ (See Ciarlet 1990, pp.63–65 for the proof)

Notice that the forumulation above uses the stress tensor in the deformed configuration where it is symmetric. 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 of which the latter is denoted with Σ. A change in the force densities due to the deformation is often ignored. These forces are then called dead loads (see Ciarlet 1990, chap.2.7).

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

Another thing that is important is that the boundary condition can and will be only prescribed on a part of the boundary.

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. However, 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 (right-) Chauchy (Green) strain tensor C and its difference from unity E is introduced. I will refer to E simply as the strain tensor. They describes the first order local change in length-scale under a deformation and are defined 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 and I is the identity tensor. In the linear theory that is used as the basis for the topology optimization, the strain E is replaced with the linearized version ε:

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

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

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

The more prominent form of which is called Hooks-law and written with the so-called stiffness tensor c:

$$\sigma = c : \varepsilon \qquad \text{or} \qquad \sigma = c\varepsilon \tag{2.2}$$

Where the following tensoroperation for rank 2 tensors is introduced that will be used in the coming sections:

$$G : V := \sum_{i,j} G_{ij} V_{ij}$$

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.

For this we first define the space $H^1_D = \{\theta \in H^1(\Omega) | \theta = 0 \text{ on } \Gamma_0 = \partial\Omega - \Gamma_g\}$

Multiplying equation 2.1b with a test function θ from this space on both sides and integrating yields:

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

Which has to hold for all test functions $\theta \in H^1_D$

Using the Greens-formula 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 vector fields $\theta : \bar{\Omega} \mapsto \mathbb{R}^3$ from H^1_D. This is also called the 'principle of virtual work' (in the reference configuration).

For very small strains the deformation gradient in front of Σ can be dropped since the diplacement gradient multiplied with the stress-tensor is of second order in ∇u. (see Braess 2013, p.280)

Furthermore, since the product of a symmetric tensor with an antisymmetric one is zero, we split $\nabla\theta$ into: $\left[\frac{1}{2}(\nabla\theta + \nabla\theta^T) + \frac{1}{2}(\nabla\theta - \nabla\theta^T)\right]$

And thus we write the equilibrium equation in the following comprehensive form:

$$\int_{\bar{\Omega}} \sigma\frac{1}{2}(\nabla\theta + \nabla\theta^T) = \int_{\bar{\Omega}} \sigma\varepsilon(\theta) = \int_{\bar{\Omega}} f \cdot \theta dx + \int_{\Gamma} g \cdot \theta da \quad \forall\theta \in H^1_D \tag{2.3}$$

For the following sections we write this as the following shorthand notation using the inner product $\langle A, B\rangle_C = \int_{\bar{\Omega}} A : CB :$

$$\langle\varepsilon(u)\varepsilon(\theta)\rangle_{C(\varphi)} = \int_{\bar{\Omega}} f \cdot \theta + \int_{\Gamma} g \cdot \theta =: F(\theta) \tag{2.4}$$

The well posedness of the weak formulation is proved using the Lax-Milgram Lemma and Korns Inequality in (Blank et al. 2014)

For an actual deformation rather than a virtual one, the functional F is called the compliance of the structure. This compliance is what is to be optimized in the following section.

2.2 Topology optimization

The topology optimization procedure used in this work is a direct implementation the works of (Blank et al. 2014) As a result of this I frequently refer to these papers and only strive to give a compact summary of the steps. Topology optimization is a relativly new technique and was first

The term topology optimization was coined in the context of optimizing mechanical structures. It is not bound to a certain implementation but to the requirement that a structure under load that is optimized via a certain functional may change its topology under that optimization. This is generally understood in the sense as to allow the nucleation of holes in a previously filled material rather than forming new strucutres in the void.

The method used throuout this work that accomplishes this uses the compliance minimization functional with a phase field description of the structure. In effect, compliance minimization maximizes the stiffness of the structure under a mass- or volume constraint while the phase field describes where material is placed in the domain.

The use of a phase-function description allows to incorporate a computationally cheap perimeter regularization via an additional term in the optimization functional. This is needed since the compliance minimization in itself is not well posed and allows high variaton in the microstructure of a part that cannot be manufactured and is not numerically stable.

2.2.1 Compliance minimization

Compliance is a very common goal function for topology optimization and defined as in equation 2.4:

$$F(u) = \int_{\bar{\Omega}} f \cdot u + \int_{\Gamma} g \cdot u$$

Here, u is the displacement solution of the mechanical system in the left-hand side of equation 2.4

However, it is still open at this time how the compliance depends on the structure of the part. This dependence is actually encoded in the Stiffness-tensor c of the mechanical system that will be constructed from the phase field after this has been rigorously defined.

2.2.2 The phase-field description and regularization energy

For the modeling of structures either a level set method or a phase field description is viable. In case of the phase-field a continuous function is chosen that here can take on values in the range from 0 to 1 where 0 represents the void and 1 the material:

$$0 < \varphi < 1$$

A penalty term is then added to the optimization to force the phasefield to condensate to either 0 or 1 depending on a forcing term from the compliance minimization. Consequently an interface forms that can expand or retract and this falls into the category of advancing front algorithms. For details see (Barles et al. 1993). The resulting equation driving the interface is an Allan-Cahn Equation as discussed in (Blank et al. 2010). However, for the general understanding the optimality conditions are of more avail.

Firstly, since integrating a value of 1 over a region, a volume constraint is easily added with a parameter m that dictates how much of the design domain is to be void:

$$\int_\Omega \varphi dx = m \cdot \mathrm{vol}(\Omega)$$

Notice that values in the intermediate range of φ contribute a lesser volume. However, these values, as they only occur in the interfacial region, will be forced to occupy a small portion of the domain.

For the reading convenience we stipulate the requirements on φ in the following space:

$$\mathcal{G}^m = \left\{ \varphi \in H^1(\Omega, \mathbb{R})) \quad \middle| \quad 0 \leq \varphi \leq 1 \quad \text{and} \quad \int_\Omega \varphi dx = m \cdot \mathrm{vol}(\Omega) \right\}$$

These requirements are later taken care of by complementary slackness and a lagrange multiplier respectively.

As stated, an additional term has to be added to the compliance functional as to force condensation of the phasefield and regularize the occurence of jumps. The term used is due to (Takezawa et al. 2010):

$$E^{\varepsilon} = \int_{\Omega} \frac{\varepsilon}{2} |\nabla \varphi|^2 + \frac{1}{\varepsilon} \Psi(\varphi) dx$$

The potential $\Psi(\varphi)$ serves as the condensation potential that forces the phase field to either 0 or 1. Here, an obstacle potential is used while for the analysis, a differentiable double-well potential is employed:

$$\Psi_{\mathrm{dw}}(\varphi) = \frac{1}{4}(\varphi^2 - \varphi)^2$$

$$\Psi_{\mathrm{obs}} = \begin{cases} (1-\varphi)\varphi, & \text{if } \varphi \in [0,1] \\ \infty & \text{else} \end{cases}$$

The two potentials are displayed in figure 2.1

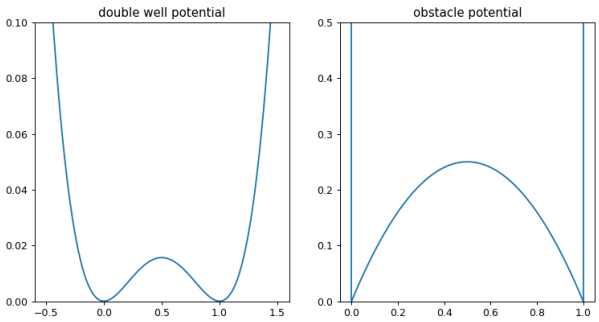

Figure 2.1: The two potentials in the Ginzburg-Landau energy term

2.2.3 Interpolation of the stiffness tensor and gravitational force

The stiffness tensor in the void is modeled by a very soft material C_{void}. For the interpolation in the interfacial region towards the material tensor C_{mat} a linear interpolation with a superimposed transition function $t(\varphi) = \varphi^3$ is used:

$$C(\varphi) = C_{\text{mat}} t(\varphi) + C_{void}(1 - t(\varphi))$$

Also, the force f occuring in the mechanical system can now be made concrete. Namely the phase-field acts as a direct scaling factor for the mass density and excludes the void from contributing any forces:

$$f = \varphi \cdot \rho_{\text{mat}} \cdot g \quad \text{with } g = 9.81 \cdot [0, 0, 1]^T$$

$$F(u, \varphi) = \int_{\bar{\Omega}} \varphi \cdot \rho_{\text{mat}} \cdot u + \int_{\Gamma_g} g \cdot u$$

2.2.4 The optimal control problem

We are now ready to state the optimality system including the first order necessary condition for a minimum.

As indicdated before, the goal is to minimize the functional made up of the compliance and the Ginzburg-Landau term:

$$\min J(u, \varphi) := E(\varphi) + F(u, \varphi) \quad \text{with } \varphi \in \mathcal{G}^m \text{ and } u \text{ forfills eq 2.4}$$

To write down the optimality conditions in a concise form, consider the control-to-state operator $S(\varphi) = u$ defined implicitly by equation 2.4. It's directional derivative $S'(\varphi)h = p$ is given by the solution to:

$$\langle \varepsilon(p)\varepsilon(\eta) \rangle_{C(\varphi)} = \langle \varepsilon(u)\varepsilon(\eta) \rangle_{C'(\varphi)h} - \int_{\bar{\Omega}} h \cdot f \cdot \eta \quad \forall \eta \in H^1_D \tag{2.5}$$

(see Blank et al. 2014 theorem 3.3)

It follows from the definition of the total differential and the chain rule that the reduced functional $\tilde{J}(\varphi) = J(S(\varphi), \varphi)$ is fréchet-differentiable with the derivative:

$$\tilde{J}'(\varphi)h = \frac{\partial}{\partial u}J(u,\varphi)p + \frac{\partial}{\partial \varphi}J(u,\varphi)h$$

Since $E(\varphi)$ is independant of u, the partial derivative with respect to u is just the right-hand side of the state equation:

$$\frac{\partial}{\partial u}J(u,\varphi)p = F(p,\varphi)$$

Which is, since $p \in H^1_D(\Omega, \mathbb{R}^d)$ is an admissible test function, equal to the left hand side of the state equation. Note that if this was not the case, an auxillary state $q \in H^1_D$ could have been introduced which would solve a system sometimes called the adjoint system as to make the following equality hold.

Now using equation 2.5 with u as a test function this can be written as:

$$\frac{\partial}{\partial u}J(u,\varphi)p = F(p,\varphi) = \langle \varepsilon(u)\varepsilon(p)\rangle_{C(\varphi)} \quad = -\langle \varepsilon(u)\varepsilon(u)\rangle_{C'(\varphi)h} - \int_{\bar{\Omega}} h \cdot f \cdot u \tag{2.6}$$

The calculation of the partial derivative of J with respect to φ is straightforward:

$$\frac{\partial}{\partial \varphi}J(u,\varphi)\xi = \varepsilon \int_\Omega \nabla\varphi : \nabla\xi + \int_\Omega \frac{1}{\varepsilon}\Psi'(\varphi)\xi dx + F(u,\xi)$$

Summing up, the reduced functional has the following directional derivative:

$$\frac{d}{d\varphi}\tilde{J}(\varphi) = 2\cdot F(u,\xi) + \int_\Omega \varepsilon\nabla\varphi\nabla\xi + \frac{1}{\varepsilon}\Psi'(\varphi)\xi \; dx - \langle \varepsilon(u)\varepsilon(\eta)\rangle_{C'(\varphi)\xi} \tag{2.7}$$

To incorporate the constraints on φ we now follow the Karush-Kush-Tucker theory. For this to work we have to make sure a constraint qualification is satisfied. Here we consider the slater condition that, for a an itermediate density material distribution in the whole domain is obviously satisfied. Thus we can assume strong duality and the complentarity follows.

To reconsider, we have the following additional requirements

$$\int_\Omega \varphi - m \; dx = 0 \tag{2.8}$$
$$\varphi - 1 \leq 0 \tag{2.9}$$
$$-\varphi \leq 0 \tag{2.10}$$

Introducing lagrange multipliers η, μ, λ the KKT-first order necessary optimality conditions then read:

$$\frac{d}{d\varphi}\tilde{J}(\tilde{\varphi})\omega + \eta \int_\Omega \omega \; dx + \mu\omega - \lambda\omega = 0 \quad \forall \omega \in H^1_D \tag{2.11}$$
$$\langle \varepsilon(\tilde{u})\varepsilon(v)\rangle_{C(\varphi)} = F(\tilde{u}, v) \quad \forall v \in H^1_D \tag{2.12}$$
$$\int_\Omega \tilde{\varphi} - m \; dx = 0 \tag{2.13}$$
$$\mu \geq 0, \quad \lambda \geq 0 \tag{2.14}$$
$$(\mu, \tilde{\varphi} - 1) = 0 \quad \text{a.e. in } \Omega \tag{2.15}$$
$$(\lambda, -\tilde{\varphi}) = 0 \quad \text{a.e. in } \Omega \tag{2.16}$$

Where the last three conditions arise due to complementarity.

$$\mathcal{L}(u, \varphi, p) = F(u, \varphi) + E^\varepsilon(\varphi) - \langle \varepsilon(u)\varepsilon(p)\rangle_{C(\varphi)} + F(p, \varphi) - \eta \int_\Omega \varphi - m \; dx$$

Notice that p is the lagrange multiplier to the mechanical system and the lagrange multiplier η is introduced to take care of the volume-contraint as

required in $\mathcal{G}^m$ We derive the Lagrange funtion formally to obtain the optimality system:

$$(\partial_u \mathcal{L})q = F(q,\varphi) - \langle \varepsilon(q)\varepsilon(\eta)\rangle_{C(\varphi)} \tag{2.17}$$

$$\begin{aligned}(\partial_\varphi \mathcal{L})\xi = 2\cdot F(u,\xi) - + \int_\Omega \varepsilon \nabla\varphi\nabla\xi + \frac{1}{\varepsilon}\Psi'(\varphi)\xi \; dx \\ - \langle \varepsilon(u)\varepsilon(\eta)\rangle_{C'(\varphi)\xi} - \eta\int_\Omega \xi dx\end{aligned} \tag{2.18}$$

We are now ready to state the complete optimality system:
The state equations:

$$\text{(SE)} \quad \begin{cases} \langle \varepsilon(u)\varepsilon(\theta)\rangle_{C(\varphi)} + F(p,\varphi) \\ \int_\Omega \varphi - m \; dx \end{cases} \tag{2.19}$$

The adjoint equation:

$$\text{(AE)} \qquad \langle \varepsilon(u)\varepsilon(\theta)\rangle_{C(\varphi)} + F(p,\varphi) \tag{2.20}$$

And the variational inequality:

$$\begin{aligned}(\partial_\varphi \mathcal{L})(\tilde{\varphi}-\varphi) &= 2\cdot F(\tilde{u},(\tilde{\varphi}-\varphi)) \\ &+ \int_\Omega \varepsilon\nabla\tilde{\varphi}\nabla(\tilde{\varphi}-\varphi) + \frac{1}{\varepsilon}\Psi'(\tilde{\varphi})(\tilde{\varphi}-\varphi)\; dx \\ &- \langle \varepsilon(\tilde{u})\varepsilon(\eta)\rangle_{C'(\tilde{\varphi})(\tilde{\varphi}-\varphi)} - \eta\int_\Omega(\tilde{\varphi}-\varphi)dx \quad \geq 0\end{aligned} \tag{VI, 2.21}$$

2.2.5 Numerial soution

2.2.5.1 Pseudo time stepping

Now, we solve for $\tilde{\varphi}$ in the optimality conditions. Equation 2.11 defines a linear functional on H^1_D which I refer to as $\nabla\mathcal{L}(\omega)$. Using a scalar product this functional can be identified with a funtion on H^1_D that we loosely call the gradient. This approach is called a gradient flow. Consequently a gradient descent in conjunction with a semi-implicit stepping scheme is used.

$$(\partial_t \varphi, \omega) = \nabla \mathcal{L}(\omega)$$

Foo:

$$\begin{aligned}(\partial_t \varphi, \omega) =& \varepsilon \int_\Omega \nabla\varphi\nabla\omega + \int_\Omega \frac{1}{\varepsilon}\Psi'(\varphi)\omega \; dx + 2 \cdot F(u,\omega) \\ & - \langle \varepsilon(u)\varepsilon(u)\rangle_{C'(\varphi)\omega} + \eta \int_\Omega \omega \; dx + \mu\omega - \lambda\omega\end{aligned} \tag{2.22}$$

We start with some initial function $\varphi^k,\ k = 0$. Inserting for $\partial_t\varphi$ its approximation $\frac{\varphi^{k+1}-\varphi^k}{\tau}$ and using φ^{k+1} for $\nabla\varphi$ we end up with:

$$\begin{aligned}\frac{1}{\tau}\int_\Omega \varphi^{k+1}\omega dx + \varepsilon \int_\Omega \nabla\varphi^{k+1} : \nabla\omega dx =& \frac{1}{\tau}\int_\Omega \varphi^k \omega dx \\ & + \frac{1}{\varepsilon}\int_\Omega \Psi'(\varphi^k)\omega dx + \langle \varepsilon(u)\varepsilon(u)\rangle_{C'(\varphi^k)\omega} \\ & - 2\int_\Omega \omega\rho_0 g u dx + \eta \int_\Omega \omega \; dx + \mu\omega - \lambda\omega\end{aligned} \tag{2.23}$$

2.2.5.2 Primal-dual active set strategy

So far we have left out the complementarity conditions in the calculation of the descent direction. For this, a primal dual active set strategy (PDAS) is used. PDAS maintains a set of active constraints for every point. A constraint is inactive if the corresponding lagrange multiplier is zero and active if φ takes on the corresponding bound. One of which has to hold due to the equations 2.15 and 2.15.

Consequently the gradient descent step is calculated with φ taking on a fixed value on the active set and being unbounded (lagrange multiplier =0) on the inactive set. Afterwards the

2.2.6 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 2.2 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.

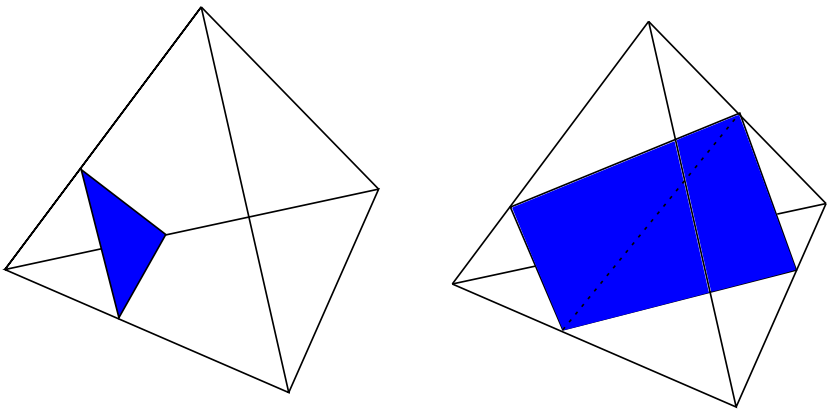

Figure 2.2: In the case of a 3D-Tetrahedra tesselation only 3 distinct cases can appear a)intersection at three edges(left) or b)intersection at 4 edges(right) or c)no intersections(not displayed)

2.3 Radial-basis-function theory

2.3.1 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.:

$$S(x_i) = F(x_i) \quad \forall i \in \Xi \tag{2.24}$$

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

By introducing the interpolation matrix A as:

$$A = \varphi(\|x_i - x_j\|)|_{i,j} \tag{2.25}$$

we can write the interpolation condition 2.24 as:

$$A\alpha = F \tag{2.26}$$

2.3.2 Existence and Uniqueness results

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

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

Where g^l means the l-th derivative of a function $g : \mathbb{R} \mapsto \mathbb{R}$. Given this property it can be shown that the interpolation matrix is always positive definite. This was first shown for the multiquadratics function and relies on the Bernstein representation theorem for monotone functions. (see Buhmann n.d., pp.11–14) for a write up or (Micchelli 1986)for the original proof. One then calls the underlying function g positive definite. A weaker requirement is that only one of the derivatives must be completely monotone. This then leads to the concept of conditonally positive definite functions in which a polynomial is added to the interpolant.

- 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

2.3.3 Commonly used Radial basis functions

I now recite some of the more often used RBFs and state if they are positive definite or conditionally so. During the course of writing the interpolation program it became clear that only local basis functions would be good candidates for a surface interpolation due to the number of vertices used in most triangular meshes and the resulting size of a dense interpolation matrix.

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

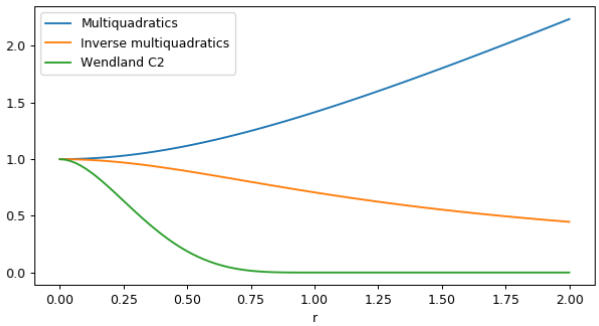

Figure 2.3: Comparison of different RBF functions. Note that a convergence to zero is not mandatory. However, the Wendland functions become zero after r=1

2.3.4 Scaling of RBF functions, ambiguities and interpolation properties

The Wendland RBF functions have a fixed spread of 1 as seen in fig. 2.3. Since spacing of the interpolation data is not fixed, a scaling of the radial argument 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 c, ie. $r' = r/c$ 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. 2.5

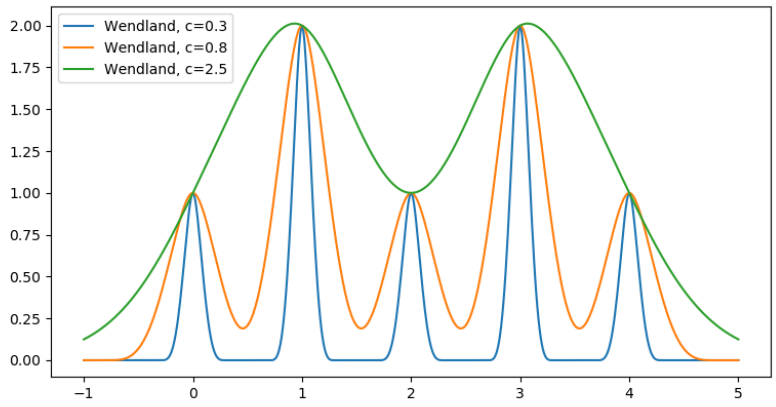

Figure 2.4: Wendland C2 functions for different scaling parameters c. The interpolation values were set to (1,2,1,2,1) at (0,1,2,3,4).

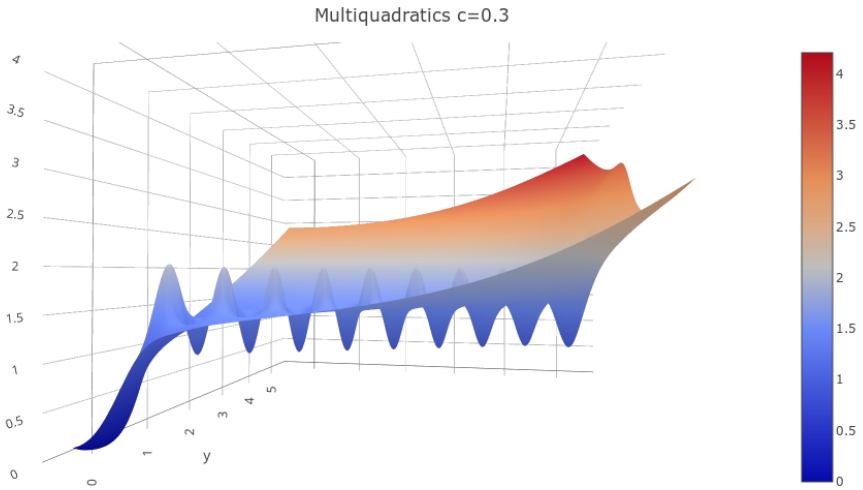

Figure 2.5: Different Radial-Basis-Funtions have different behaviours for off-site values. Multiquadratics grow toward infinity. Displayed is a 1-2 comb in two dimensions

2.3.5 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:

$$S(\mathbf{x}_i + \epsilon \mathbf{n}_i) = F(\mathbf{x}_i + \epsilon \mathbf{n}_i) = \epsilon \tag{2.27}$$

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.

2.4 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.

2.4.1 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. 2.7 and 2.8 To avoid having to project a new midpoint to the surface, the two vertices of the edge are joined at either one of them.

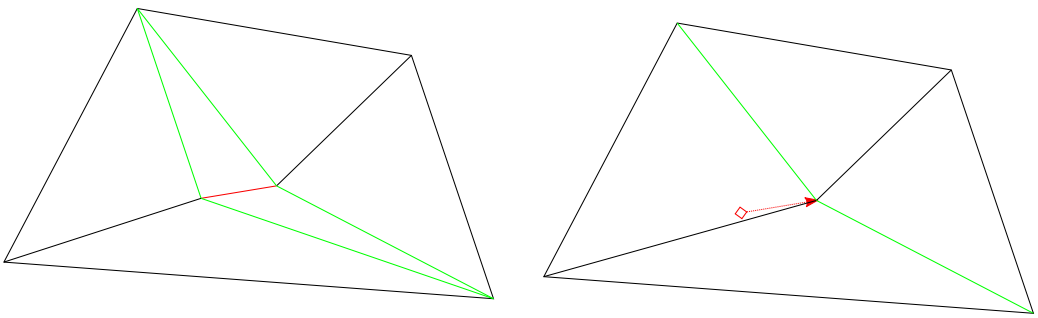

Figure 2.6: Edge collapse with the new point at one of the endpoints

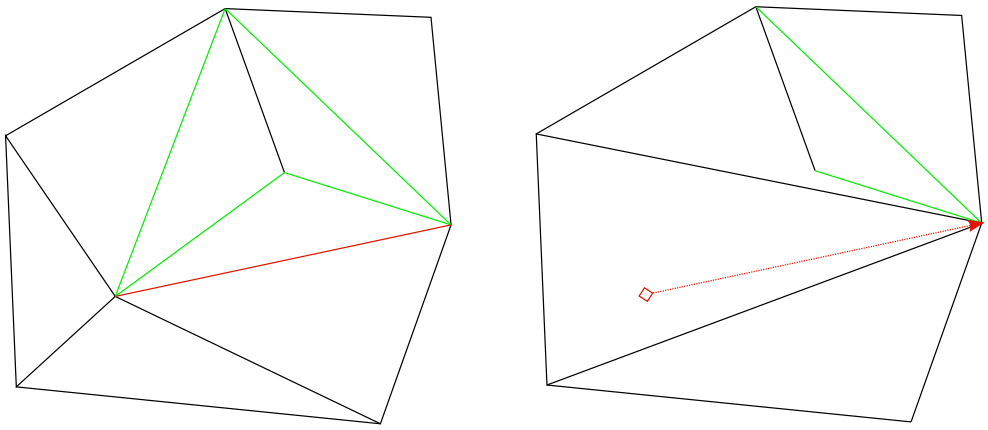

Figure 2.7: Illegal edge collapse with more than two common neighbors for the edges endpoints

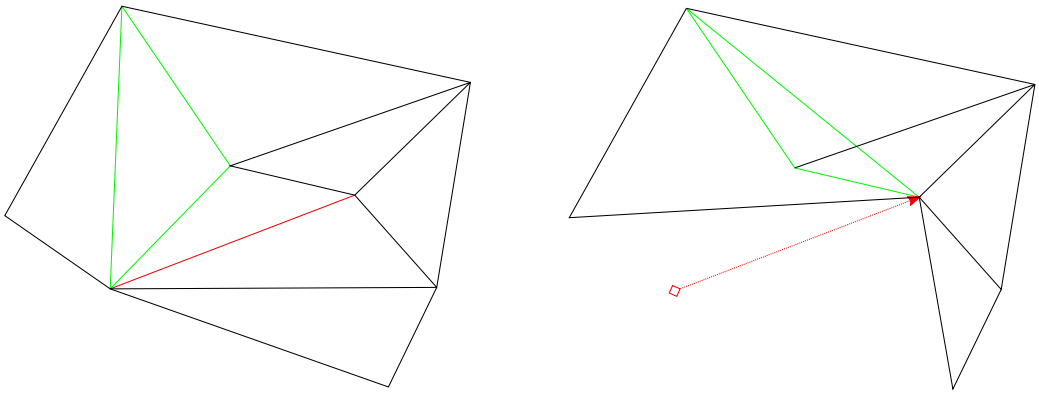

Figure 2.8: Illegal edge collapse with a triangle flip

2.4.2 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.

2.4.3 Edge flip

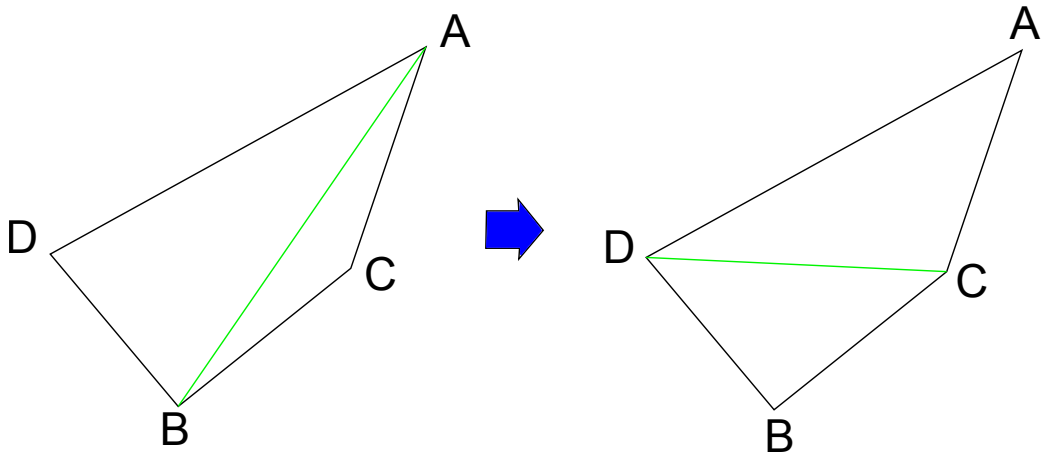

Figure 2.9: Edge flip

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:

2.4.4 Vertex smoothing

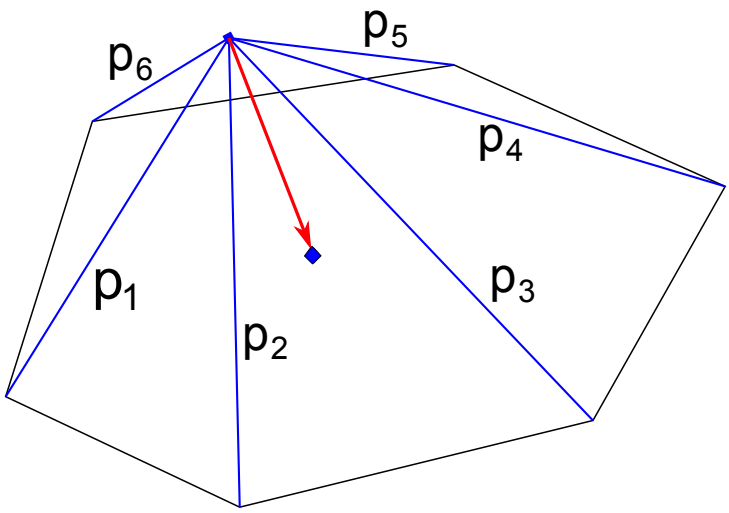

Figure 2.10: Vertex smoothing

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 2.11 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.

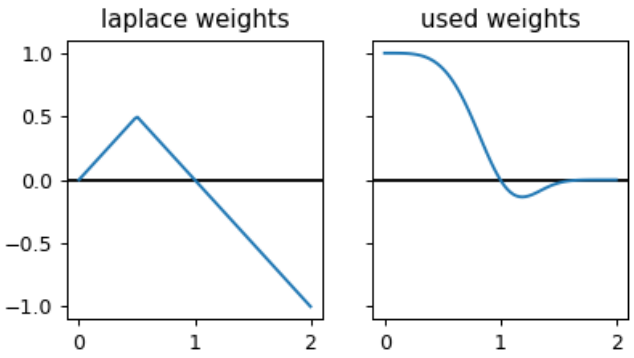

Figure 2.11: The weight function used compared to the laplace weights

2.4.5 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 iterations with a fixed steplength of one. This has proven much faster convergence than the exact steplength. This is due to the fact that around the surface the slope of the function is one by construction.

The descent algorithm reads as follows

Algorithm 1: Project a vertex x_0 onto the surface

Input : x_0 and *eps*

```
1  i=0
2  while i < steplimit do
3  |   if i mod 3==0 then
4  |   |   calculate ∇f(x_i)
5  |   calculate f(x_i)
6  |   if  f(x_i) < eps then
7  |   |   return x_i
8  |   stepsize = f / ||∇f(x_i)||
9  |   clip stepsize
10 |   x_{i+1} = x − stepsize · ∇f(x_i)
11 end
```

2.5 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

3.1 interpolation

The vertices of the extracted isosurface are the principal points for the interpolation.

As stated in sec. 2.3.5, additionally, the off-surface values both in the positive as well as the negative direction are incorporated into the interpolation. These are calculated via the original meshes vertices v and triangle normals n_T as follows:

$$\mathbf{v}_{\text{off}} = \mathbf{v} \pm \varepsilon \frac{\mathbf{n_v}}{\|\mathbf{n_v}\|}$$

where $\mathbf{n_v}$ is assembled of the triangles $\mathcal{N}_T$ containing v as a vertex:

$$\mathbf{n_v} = \sum_{T \in \mathcal{N}_T} \frac{1}{\|\mathbf{v} - \mathbf{v}^{\mathbf{T}}_{\text{cent}}\|} \mathbf{n}_T$$

Here $\mathbf{v}^{\mathbf{T}}_{\text{cent}}$ is the centroid of the triangle.

Since the normals of the triangles point outward of the structure as in the

stl specification we have higher interpolant values outside the object and a gradient pointing outwards.

Experimentation with ε only yielded surfaces that were vertiable for projection for small values. I settled with a general forumla setting ε as the average between the longest edge in the mesh and the smallest edge divided by 10, ie.:

$$\varepsilon = \frac{e_{\text{longest}} + e_{\text{shortest}}}{20}$$

The scale factor of the rbf functions was set to 2.5 times the longest edge in the mesh where lower values would disrupt the contiguous surface.

3.2 surface conditioning

A principal problem is that of the values of the interpolant in between the datasites. In the case of an implicitly defined surface this influences not only the shape of the zero level set but also the slope around the border of the surface. As an illustration take a look at fig. 3.1

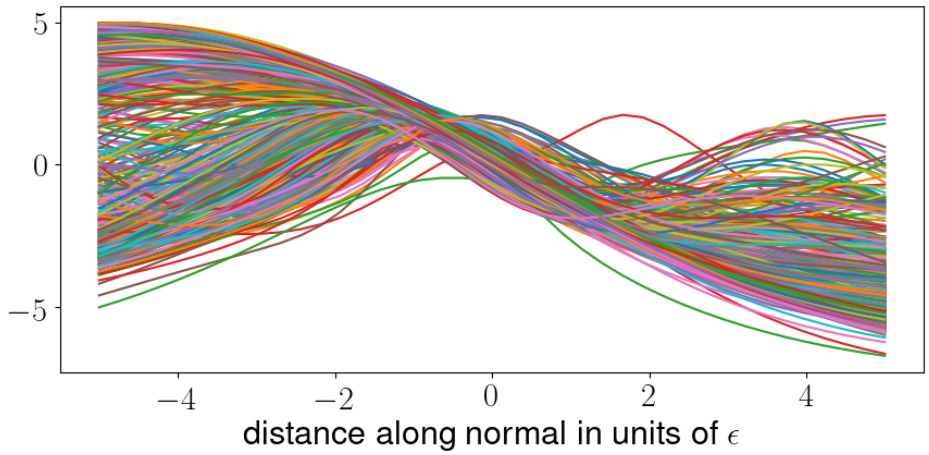

Figure 3.1: Interpolant values along the normal direction sampled at the triangle centroids

The interpolation matrix 2.25 is then constructed as a sparse matrix with a

scale factor c as explained in sec. 2.3.4 and the system 2.26 is subsequently solved for the coefficients.

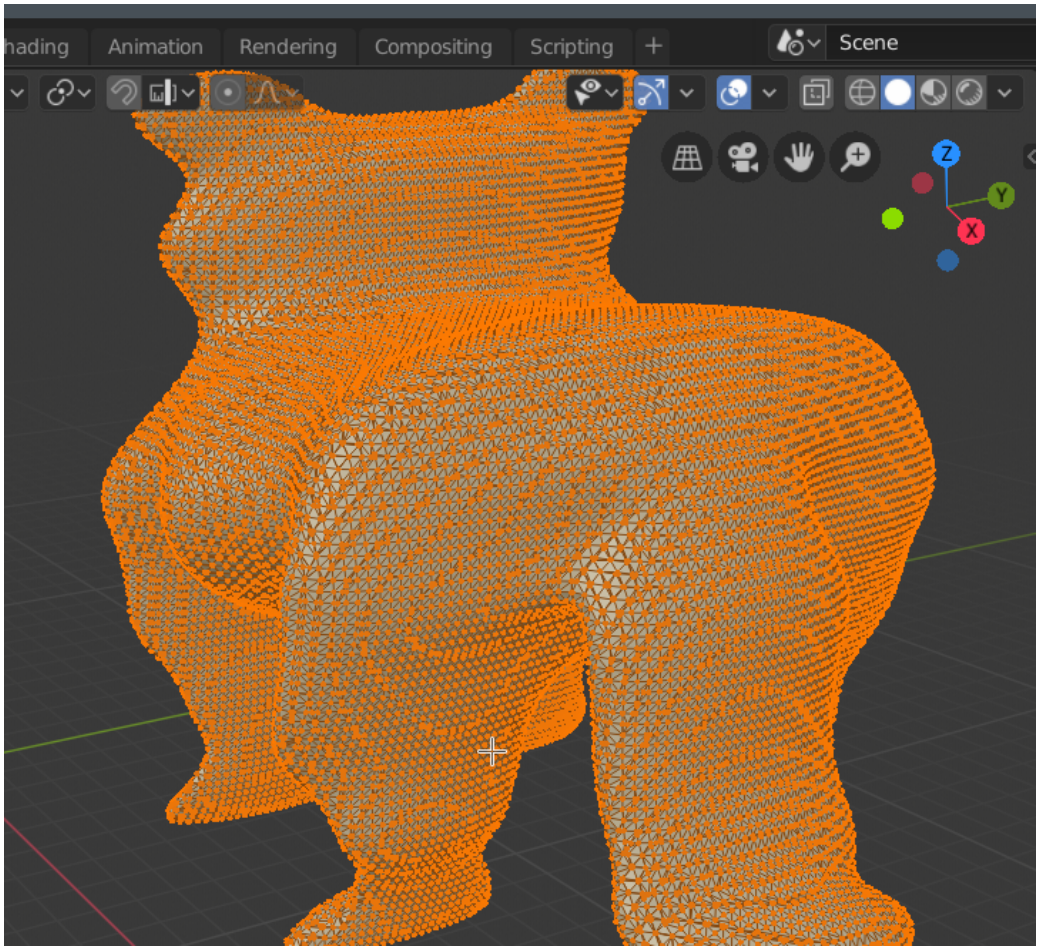

Figure 3.2: A testmodel interpolated with the Wendland functions

3.3 Smoothing

The resulting isosurfaces of the topology optimization generally had rough surfaces with seemingly random small surface features like bumbs dents. Since an interpolation rather than a fit were used, these features would be preserved and result in small refinement for those features.

To circumvent this, a mesh smoothing method was employed to smooth the isosruface mesh. Since the often used laplace smoothing diminishes volume I opted for taubin smoothing as described in (Taubin 1995).

The smoothing is very similar to the introduced vertex-smoothing above. For a vertex v_i and neighbors v_j, the position of the vertex is shifted with a weighed average of the neigbors positions:

$$\Delta v_i = \sum_{j \in \mathcal{N}_i} w_{ij}(v_j - v_i)$$

With $w_{ij} = 1/|\mathcal{N}_i|$ as the number of neighbors. The shift is then added partially to the original vertex.

$$v_i' = v_i + \lambda \Delta v_i \qquad 0 < \lambda < 1$$

For taubin smoothing an analogous second smoothing step is introduced with a negative λ ie. an expansion. The coefficient for this is denoted μ with the restriction that $0 < \lambda < -\mu$. In (Taubin 1995) taubin shows that this acts as a lowpass filter and prevents shrinkage of the model.

For the application I used values of $\lambda = 0.40$, $\mu = -0.52$ and made 40 smoothing iterations.

3.4 Projection step

Inserting a new vertex in an edge split and smoothing a vertex requires a projection of a midpoint onto the surface. For sufficiently small distances to the surface this projection is assumed to be orthogonal since the gradient has no components in the tangent plane to the isosurface. This is however no requirement since vertex smoothing will take care of adjusting the positions of the vertices on the surface later on.

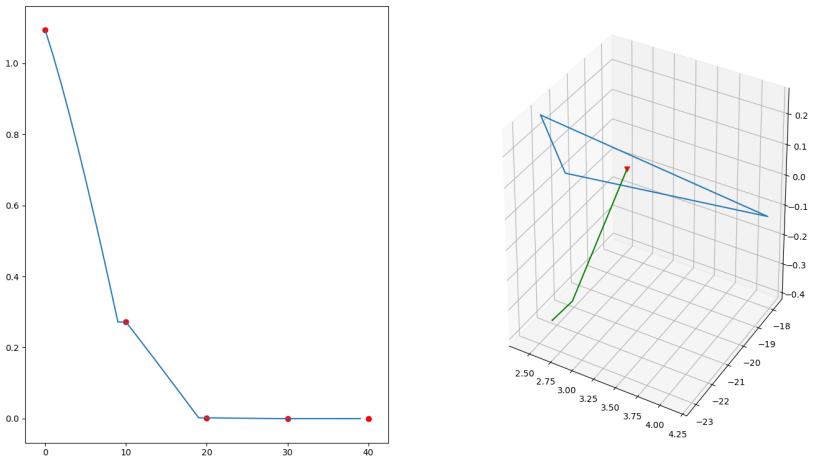

Figure 3.3: Fast convergence of the gradient descent onto the surface for a steplength of 1

3.5 Remeshing

Generally, points on the boundary are not touched since they form a fixed interface to

The remeshing algorithm is taken from (Dassi et al. 2016)

Algorithm 2: The remeshing procedure

Input : Target edge length l_{6d} and σ for HDE

```
i=0
while i <maxiter do
    smalledges={l^e_6d < 0.5 · l_6d for e in edges }
    j=0
    while smalledges and j < 10 do
        collapse smalledges
        smooth random 30% of vertices
        flip all edges
        update smalledges
        j+=1
    end
    longedges={l^e_6d > 1.5 · l_6d for e in edges }
    split long edges
    flip all edges; smooth all vertices
    flip all edges
end
```

3.5.1 CONVERGENCE ANALYSIS

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)

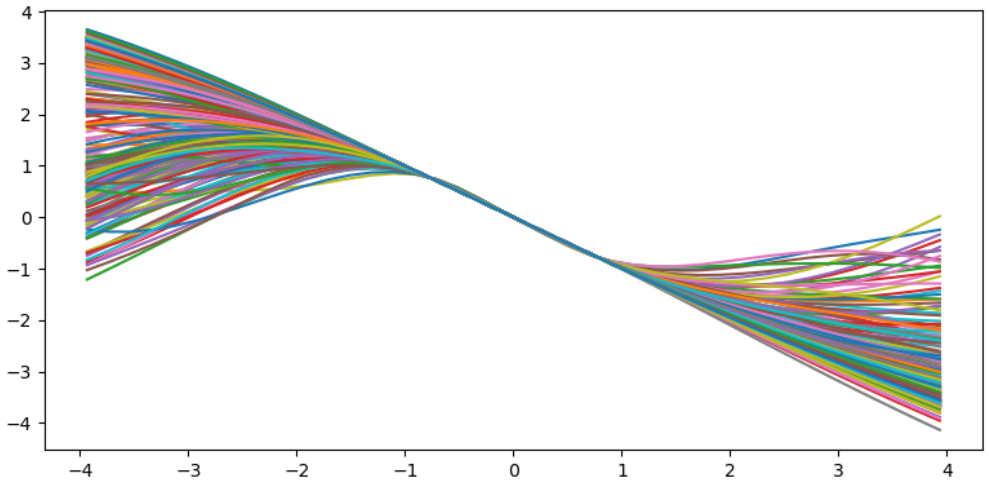

Figure 4.1: At the vertices of the mesh in between the offset interpolation points the RBF-interpolant is well behaved. The offset interpolation points are located at $\pm$ 0.78 in units of the actual mesh

4.1 The topology optimization models

4.1.1 The bridge

- parameters: gamma, eps
- domain description and boundary conditions
- how many tetrahedra

4.1.2 The table

4.1.3 The tower

4.2 Analysis of the results and problem

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

Barles, G., Soner, H. & Souganidis, P., 1993. Front propagation and phase field theory. *SIAM Journal on Control and Optimization*, 31(2), pp.439–469. Available at: https://epubs.siam.org/doi/abs/10.1137/0331021 [Accessed October 5, 2019].

Blank, L. et al., 2014. Relating phase field and sharp interface approaches to structural topology optimization. *Control, Optimisation and Calculus of Variations (ESAIM-COCV)*, 20, pp.1025–1058. Available at: https://epub.uni-regensburg.de/34578/ [Accessed October 5, 2019].

Blank, L. et al., 2010. Phase-field approaches to structural topology optimization. Available at: https://epub.uni-regensburg.de/14656/ [Accessed October 5, 2019].

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

Braess, D., 2013. *Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie* 5., überarb. Aufl., Berlin: Springer Spektrum.

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

Ciarlet, P., 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].

Dassi, F., Farrell, P. & Si, H., 2016. A novel surface remeshing scheme via higher dimensional embedding and radial basis functions.

Micchelli, C.A., 1986. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2(1), pp.11–22. Available at: https://doi.org/10.1007/BF01893414 [Accessed October 15, 2019].

Takezawa, A., Nishiwaki, S. & Kitamura, M., 2010. Shape and topology optimization based on the phase field method and sensitivity analysis. *Journal of Computational Physics*, 229(7), pp.2697–2718. Available at: http://www.sciencedirect.com/science/article/pii/S0021999109006962 [Accessed October 5, 2019].

Taubin, G., 1995. Curve and surface smoothing without shrinkage. In *Proceedings of IEEE international conference on computer vision*. Proceedings of IEEE international conference on computer vision. pp. 852–857.

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