



Variational Mesh Optimisation







- 2 Computation Of Inverse Trace Operators
- 3 Nonlinear Mesh Quality Functionals
- 4 r-Adaptivity
- 5 Mesh Alignment
- 6 Computational Aspects







Let $\Omega \subset \mathbb{R}^d$, d = 2, 3 be the region of interest, Ω_h its polygonal bounded approximation and $\mathcal{T}_h = \bigcup_{i=0}^N K_i$ be a regular, conforming discretisation of Ω_h into hypercubes or simplices. On this, we want to apply the Finite Element Method to solve some kind of PDE. For this, we want to optimise the (potentially moving) given mesh according to some obvious criteria like

- Equidistribution of the domain's volume over all cells, or concentration of cells according to some criteria,
- 2 Maximisation of the minimum angle over all cells,
- 3 Maintaining the regularity and connectivity of a moving mesh
- 4 Without adding any vertices or cells.





Example 1: Basics

 $\Omega = [0,1]^2$. According to which criteria do we move the vertex v?





Some Classes Of Mesh Optimisation Methods

- 1 Algebraic and heuristic methods (i.e. graph-based Laplacian smoothing):
 - + Easy to implement and often cheap, successful for special purposes.
 - No guarantees on mesh quality or convergence behaviour. Not general purpose, often destroy local (*h*-)adaptivity.
- **2** PDE-based (i.e. energy minimisation-based) methods:
 - + For every set of requirements, there exists a method. Very adaptable, often existing discretisation methods and solvers can be re-used.
 - More sophisticated methods often require the solution of nonlinear PDEs, while cheaper methods do not match the above criteria.

Jordi Paul, 03.03.2016



Optimal Deformations In A Variational Context

That means we are looking for an optimal deformation Φ^* that minimises a functional \mathcal{F} over a set V of admissible deformations. Let $\mathcal{V} := \mathcal{V}(\mathcal{T}_h)$ be the vertices of \mathcal{T}_h . Then we have

$$\begin{split} \Phi^* &= & \operatorname{argmin}_{\Phi \in D} \mathcal{F}(\Phi) \\ D &:= \left\{ \Phi : \mathcal{T}_h \to \mathbb{R}^d : \Phi \in \mathcal{C}^0(\mathcal{T}_h), \\ & \forall K \in \mathcal{T}_h : \forall x \in K : \nabla \Phi_{|K}(x) \in SL_d, \\ & \forall v \in \mathcal{V} : v \in \partial\Omega \Rightarrow \Phi(v) \in \partial\Omega \right\} \end{split}$$

(In practice, we need to use *Optimal Variations* instead, see [Rum96].) The big advantage: We can use the same FEM tools as we do for the original PDE if we discretise the space D to some D_h accordingly.

Jordi Paul, 03.03.2016



Examples Of Functionals Leading To Linear Variational Problems

Motivational, in strong form, assuming sufficient regularity:

$$\mathcal{F}(\Phi) = \|f - \Delta \Phi\|_{L^2(\mathcal{T}_h)}$$

(minimisation of harmonic energy)

2 $\mathcal{F}(\Phi) = \|f - \nabla \cdot (\lambda \nabla \cdot \Phi + \mu \frac{1}{2} (\nabla \Phi + (\nabla \Phi)^T)\|_{L^2(\mathcal{T}_h)}$ (minimisation of linear elastic energy, assuming homogeneous isotropic material with Lamé coefficients λ, μ .)

3
$$\mathcal{F}(\Phi) = \|f - \Delta^2 \Phi\|_{L^2(\mathcal{T}_h)}$$

(minimisation of biharmonic energy)

f is some function of sufficient regularity, in many applications $f \equiv 0$. In almost all cases, $D_h \subset \mathbb{P}_1(\mathcal{T}_h)$ or $D_h \subset \mathbb{Q}_1(\mathcal{T}_h)$. Which of the criteria (1) - (4) can be fulfilled or implemented?

Jordi Paul, 03.03.2016









Assume now that the domain is moving: $\Omega = \Omega(t) \forall t \in [0, \overline{t}]$. Let $\Omega_0 =: \Omega(0)$ be given, but at each instant t only the position of the boundary $\partial \Omega(t)$ is known. That means we are looking for

$$\varphi: [0,\overline{t}] \times \Omega_0 \to \mathbb{R}^d, \Omega(t) := \varphi(t,\Omega_0),$$

but only

$$\mathrm{tr}\varphi:[0,\overline{t}]\times\partial\Omega_0\to\mathbb{R}^d$$

is known. So we have to find an inverse trace operator that extends the boundary movement into the interior. In general, ${\rm tr}\varphi$ is unknown and part of the solution to the original PDE (like the position of the free capillary boundary, the evolution of the phase boundary etc.).

Jordi Paul, 03.03.2016





Example 2: Moving Upper Boundary



Figure: Left: Ω_0 , right: $\Omega(t_{50})$ with interior mesh computed by a hyperelasticity based mesh optimiser.

Jordi Paul, 03.03.2016



Example 2

Set $\Omega_0 := [0,1]^2$, $0 = t_0 < \cdots < t_{50} = \overline{t} = 0.5$ and let

$$\forall x \in \partial \Omega_0 : \varphi_{\Gamma}(t, x) = \begin{cases} (x_1, x_2 + \frac{1}{2}tx_2\sin(2\pi x_1))^{\mathsf{T}}, & x_2 = 1\\ (x_1, x_2)^{\mathsf{T}}, & \text{else} \end{cases}$$

For a given discretisation \mathcal{T}_h of a reference domain $\hat{\Omega}_t$ (assume for now $\hat{\Omega}_t = \Omega_0$), define

$$V(t) = \left\{ v \in \mathbb{P}_1(\hat{\Omega}_t) : v_{|\partial\hat{\Omega}_t} = \varphi_{\Gamma}(t) \right\}, W(t) = \left\{ w \in \mathbb{P}_1(\hat{\Omega}_t) : w_{|\partial\hat{\Omega}_t} = 0 \right\}$$

$$orall k=1,\ldots, {\sf N}:$$
 Compute $\Omega(t_k)=arphi(t_k,\hat\Omega_t)$ by finding $\Phi_h\in V(t):$

$$\forall \Psi_h \in W(t_k) : \frac{\partial \mathcal{F}(\Phi_h)}{\partial \Psi_h} = 0 \text{ and setting } \Omega(t_k) = \Phi(\hat{\Omega}_t).$$
Jordi Paul, 03.03.2016 Computation Of Inverse Trace Operators





Example 2: Harmonic Energy Minimisation With Fixed Reference Domain

Let
$$\hat{\Omega}_t = \Omega_0$$
 and $f \equiv 0$.
 $\frac{\partial \mathcal{F}(\Phi_h)}{\partial \Psi_h} = (\nabla \Phi_h : \nabla \Psi_h)_{L^2(\hat{\Omega}_0)}.$

Because the domain is not convex, the solution is not injective, resulting in a non-regular mesh (called *mesh tangling* in literature). (Better idea: Minimise the energy of Φ_h^{-1} , but this leads to a nonlinear system.)







Example 2: Harmonic Energy Minimisation With Moving Reference Domain

As an approximation to using the inverse mapping Φ_h^{-1} : Let $\forall k : \hat{\Omega}_{t_k} := \Omega(t_{k-1})$ and $f \equiv 0$.

$$\frac{\partial \mathcal{F}(\Phi_h)}{\partial \Psi_h} = \left(\nabla \Phi_h : \nabla \Psi_h\right)_{L^2(\hat{\Omega}_{t_k})}.$$

No more *mesh tangling*, but no good lower bound on the angles. Even without exploring items (1) and (2), this shows some limitations.









Example 2: Comparison Of Angles

(a) Worst angles for various mesh quality functionals.

(b) $\Omega_{t_{50}}$ for the hyperelasticity based functional.

Jordi Paul, 03.03.2016



Why We Should Still Use Linear Variational Mesh Optimisation

Advantages:

- Cheap
- Sufficient in many cases
- Recycling of FE knowledge
- Can easily respect local (h-)adaptivity
- Some drawbacks can be overcome (like extending it for equidistribution or *r*-adaptivity)

Disadvantages:

- Lack of robustness
- No direct control of mesh quality
- There exist cases where it is NOT sufficient
- Overcoming the drawbacks leads to nonlinear problems, offsetting some of the advantages

Additional keywords: Monitor functions, continuation methods, shear stiffening. [GKT10, HR11]

Jordi Paul, 03.03.2016







Deriving A Class Of Functionals

So far, the suitability of a functional for measuring mesh quality was heuristic and coincidental. We now approach optimal meshes in a more structured fashion and state our basic assumptions on a general mesh quality functional (see [Rum96]).

- **1** Assumption: $\forall K \in \mathcal{T}_h : \exists \hat{K} \text{ of optimal shape.}$
- **2** Axiom: $\mathcal{F}(\Phi) = \sum_{K \in \mathcal{T}_h} \mu_K \mathcal{F}_h(K, \Phi)$ (Locality).
- Assumption: ∀c ∈ ℝ^d : F(K, Φ + c) = F(K, Φ) (Translation invariance).
 ⇒ ∀K ∈ T_h : ∃F̃ : SL_d → ℝ : F(K, Φ) = F̃(K, ∇Φ).

As before, Φ should be regular (meaning det $\nabla \Phi > 0$).

Jordi Paul, 03.03.2016





Reference Mappings

Let $R_{\mathcal{K}}: \hat{\mathcal{K}} \to \mathcal{K}$ be the reference mapping which uniquely maps $\hat{\mathcal{K}} \mapsto \mathcal{K}$.

$$R_{\mathcal{K}}(\Phi) := \Phi \circ R_{\mathcal{K}}$$

$$\Rightarrow \mathcal{F}(\mathcal{K}, \Phi) = \mathcal{F}(R_{\mathcal{K}}(\Phi))$$

$$\Rightarrow \mathcal{F}(\mathcal{K}, \Phi) = \tilde{\mathcal{F}}(\nabla R_{\mathcal{K}}(\Phi))$$

Furthermore:

4 Assumption: ∀_{Q∈SOd} F̃(∇R_K(Φ)) = F̃(Q∇R_K(Φ)) (Frame indifference, implies translation invariance).
 5 Assumption: ∀_{Q∈SOd} F̃(∇R_K(Φ)) = F̃(∇R_K(Φ)Q) (Isotropy).





Rivlin-Erikson Representation Theorem (see [Cia88]) gives the existence of

$$\tilde{F}(\nabla R_{\mathcal{K}}(\Phi)) = F(\|\nabla R_{\mathcal{K}}(\Phi)\|_{F}^{2}, \|\mathrm{Cof}\nabla R_{\mathcal{K}}(\Phi)\|_{F}^{2}, \det \nabla R_{\mathcal{K}}(\Phi)).$$

(the principal invariants of a $\mathbb{R}^{3\times 3}$) matrix. Note that for $d = 2, \|\nabla R_{\mathcal{K}}(\Phi)\|_{F}^{2} = \|\mathrm{Cof}\nabla R_{\mathcal{K}}(\Phi)\|_{F}^{2}$).

$$\Rightarrow \mathcal{F}(\Phi) = \sum_{K \in \mathcal{T}_h} \mu_K F(\|\nabla R_K(\Phi)\|_F^2, \|\operatorname{Cof} \nabla R_K(\Phi)\|_F^2, \det \nabla R_K(\Phi))$$

$$Iim_{s\to 0} F(\cdot, \cdot, s) = \infty \text{ (Regularity property).}$$

This already means that ${\mathcal F}$ cannot be convex, but has to be at least polyconvex.

Jordi Paul, 03.03.2016



[Rum96, Theorem 1]:

If \mathcal{F} is polyconvex, $\sum_{K \in \mathcal{T}_h} \mu_k = 1$, $\mathcal{F}(Id) < \infty$ and F satisfies the regularity property,

- (a) then there exists an optimal variation Φ^*
- (b) which is globally injective iff it has no self-intersections at the boundary.
- (c) For every minimising sequence there exists a subsequence that converges to an optimal variation.
- Φ^* is not unique in general.

This functional belongs to a class of stored-energy functionals for isotropic hyperelastic materials, see [Cia88]. Note that the original theorem requires Φ to be piecewise linear, but can be extended [HR11]. Jordi Paul, 03.03.2016 Nonlinear Mesh Quality Functionals 17/37



Geometric Interpretation And An Example

- **1** $\|\nabla R_{\mathcal{K}}(\Phi)\|_{F}^{2}$ measures the edge length derivation.
- **2** $\|\operatorname{Cof} \nabla R_{\mathcal{K}}(\Phi)\|_{F}^{2}$ measures the facet deformation.
- **3** det $\nabla R_{\mathcal{K}}(\Phi)$) measures the volume change.

Example for a local functional:

$$\tilde{F}(\nabla R_{K}(\Phi)) = c_{f} \int_{K} (\|\nabla R_{K}(\Phi)\|_{F}^{2} - d)^{2} dx + \int_{K} (\det \nabla R_{K}(\Phi))^{p_{d}} dx + \int_{K} \frac{c_{d}}{\left(\det \nabla R_{K}(\Phi)\right) + \sqrt{\delta_{r}^{2} + \left(\det \nabla R_{K}(\Phi)\right)^{2}}\right)^{p_{d}} dx$$

The structure of the local functional determines the weighting of angles, shape deformation and volume change. Jordi Paul, 03.03.2016 Nonlinear Mesh Quality Functionals 18/37





Important Questions

- Does the frame independence rule out local effects?
- 2 What if anisotropic meshes are desired?
- 3 What is a reference cell?





Important Questions

- Does the frame independence rule out local effects?
- 2 What if anisotropic meshes are desired?
- 3 What is a reference cell?



Figure: *r*-adaptivity: Condensing a mesh near an implicitly given surface.





Reference Cells, Isotropic Case

In general: \hat{K} is different for every cell K, but every \hat{K} is a rotated, scaled (and translated) *normalised* reference cell, i.e.

Hypercubes: $\hat{K}_n = [-1, 1]^d$



Simplices:







Optimal Scales, Isotropic Case Denote $R_{K}(\Phi) = \frac{R_{K,n}(\Phi)}{h(K)}$. Now we have $\forall K \in \mathcal{T}_{h} : \hat{K} = h(K)\hat{K}_{n}$. How to find h(K)? (*Id* is a local minimiser and conservation of volume)

$$\Rightarrow h(K) = \sqrt[d]{\lambda(K)} \sum_{k \in \mathcal{T}_h} \det \nabla R_{K,n}(Id)$$

where $\sum_{K \in \mathcal{T}_h} \lambda(K) = 1$

and the weights λ can be chosen according to special requirements, i.e. according to some kind of concentration function

$$c: \mathcal{T}_h \to \mathbb{R}, \lambda(K) = rac{c(K)}{\sum_{K \in \mathcal{T}_h} c(K)}.$$

Jordi Paul, 03.03.2016



Choosing The Weights λ

Possible choices for the concentration function:

- **1** $c \equiv const$: Equidistribution.
- 2 c(K) = vol(K): Preservation of cell volume.
- 3 $c(K) = b^{l(K)}$, where b is the refinement base and l(K) the refinement level.
- c(K) = f(dist(s_{Φ*(K)}, Γ)) for some set Γ, like the boundary or a surface.
- 5 $c(K) = g(\eta(K))$, where $\eta(K)$ is an a posteriori error estimate for the residual.

But: This does make the problem to solve any easier.

Jordi Paul, 03.03.2016







Different Types Of Local Adaptivity

Common types of local adaptivity:

- **1** *h*-adaptivity: Locally refine cells, creating more DoFs.
- **2** *p*-adaptivity: Locally use a different FE basis.
- 3 *r*-adaptivity: Move mesh vertices.

r-adaptivity has the advantage that it does not increase the number of DoFs or modify the adjacency structure of the underlying FE spaces, although the cell sizes cannot be as quickly varying as with *h*-adaptivity. *r*-adaptivity can easily be realised by choosing a suitable concentration function $c = c(K), K \in \mathcal{T}_h$.





Example 3: Excentrically Rotating Screws



Figure: Excentrically rotating screws with different angular velocities. Geometry courtesy of O. Mierka.

 Different angular velocities lead to large mesh deformations if the boundary vertices are not allowed to move within surfaces.

- Most linear variational methods cannot deal with slip boundary conditions.
- Special purpose code can compute rotations of about 30°.

Jordi Paul, 03.03.2016





Here: Chose c(K) = f(|gapwidth|) to allow cell to drastically change their size. Using slip boundary conditions can improve the angles, but reduce boundary approximation quality. Functional fine tuning is essential.



Figure: Red: Mesh with two slip boundaries. Black: Pinned vertices at the outer boundary



Figure: Worst angle in the mesh over time.



fakultät für **m**athematik



Jordi Paul, 03.03.2016





Mesh Alignment



Interior boundaries like a phase boundary in two phase flow or the liquid-solid boundary in fluid-structure interactions can be represented in different ways, including:

- Implicit (i.e. levelset) based representation
 - + Allows changes to the topology, fixed reference mesh.
 - Only \sqrt{h} accuracy in many error estimates, interface 2 reconstruction needed
- Sharp interface representation
 - + When using isogeometric FE, accuracy of h^p is possible in many error estimates. Easy ALE formulation.
 - No topology changes possible. Needs very robust mesh deformation methods.

Idea by S. Basting: Incorporate mesh alignment in the mesh quality functional. [BW13]

Jordi Paul, 03.03.2016

Mesh Alignment





Defining The Penalty Term

Let $\mathcal{H} : \mathbb{R} \to [0,1]$ be a \mathcal{C}^1 regularisation of the Heaviside function and $l : \Omega_h \to \mathbb{R}$ and $l_0 = \{x \in \Omega_h : l(x) = 0\}$ be the implicit representation of the surface $\Gamma_h \subset \Omega_h$. Define the local vertex sets $v_{K,i}$ and penalty term

$$\forall K \in \mathcal{T}_h : P(K) = \sum_{v_{K,i} \neq v_{K,j}} \mathcal{H}(-l(v_{K,i})l(v_{K,j})).$$

It is easy to see that

$$\forall K \in \mathcal{T}_h : P(K) = 0 \iff I_{|K} \ge 0 \lor I_{|K} \le 0.$$

I does not even have to be a distance function.

Jordi Paul, 03.03.2016

Mesh Alignment







Figure: Mesh aligned with implicit surface. Left: t = 0, right: t = 0.02.

The set of points, edges and faces (in 3d) aligned with the mesh is not fixed and can vary whenever the underlying problem changes, i.e. the implicit surface or the mesh moves. Jordi Paul, 03.03.2016 Mesh Alignment 29/37







Figure: Right: Topological problem for hypercube meshes, left: Combining mesh alignment and *r*-adaptivity.

In general, the topological problem for hypercubes could be overcome by

- not penalising diagonal cuts,
- using special, possibly locally refined meshes or
- **3** only approximately fulfilling the alignment condition. Jordi Paul, 03.03.2016 Mesh Alignment





Computational Aspects

The following characteristics of the functional make it hard to treat numerically:

- **1** Polyconvexity with many local minimisers.
- 2 Highly nonlinear, containing at the very least rational functions.
- 3 The gradient of the alignment condition vanishes in all local minimisers, meaning the most sophisticated applicable method for constrained optimisation is the quadratic penalty method.
- 4 The term $(\|\nabla R_{K,n}\|_F^2 d)$ is needed for coercivity but already means $\operatorname{cond} \mathcal{F}(\Phi) = O(h^{-2})$.

All this means the computational cost can become prohibitive.

Computational Aspects

32/37

Solver Details: Two different approaches Variational derivatives (see Appendix I) with Newton-Krylov solvers:

- Works well for computing extension operators using non-moving reference domains.
- Becomes unstable for non-uniformly discretised reference domains.
- Newton fails to converge for $\lambda(K) \neq \text{const.}$
- No variational formulation of the alignment condition.

Expressing $F(\nabla R_{K,n})$ directly as functional of the vertex coordinates $x_{i,j}$ of \mathcal{T}_h , using tools from nonlinear optimisation:

- A Newton solver with approximate Hessian tends to converge to very near local minima (S. Basting)
- Nonlinear CG with strong Wolfe linesearch works very well, but difficult to precondition.



technische universität dortmund







The hyperelastiticy based method is set to conserve the volume distribution of the initial configuration, before the boundary gets transformed to the unit circle.

| Level | D(u): D(v) | $c_{f} = 1, p_{d} = 1$ | $c_{f} = 1, p_{d} = 2$ | $c_{f} = \frac{1}{100}, p_{d} = 1$ | $c_f = \frac{1}{100}, p_d = 2$ |
|-------|------------|------------------------|------------------------|------------------------------------|--------------------------------|
| 3 | 271 | 266 | 241 | 600 | 778 |
| 4 | 603 | 609 | 548 | 1348 | 1800 |
| 5 | 1256 | 1348 | 1207 | 2991 | 4307 |
| 6 | 2718 | 3131 | 2745 | 7548 | 9914 |

Table: Mean of the number of iterations over 50 timesteps.

Jordi Paul, 03.03.2016

Computational Aspects





Iteration Numbers For Example 3

Mean value of the number of solver iterations over the first 2000 timesteps. In all cases $p_d = 2$ and slip BCs on the inner boundary. Different BCs on the outer boundary.

| | | | slip | slip | dirichlet | dirichlet | |
|-------|-------|-------|-------------------|------------------|-------------------|-------------------|--|
| Level | Cells | DoFs | $Q_1, c_f = 0.01$ | $Q_1, c_f = 0.1$ | $Q_1, c_f = 0.01$ | $Q_1, c_f = 0.11$ | |
| 0 | 720 | 2160 | 78 | 86 | 142 | 153 | |
| 1 | 2880 | 7200 | 654 | 169 | 660 | 528 | |
| 2 | 11520 | 25920 | 1498 | 380 | 1461 | 821 | |
| Level | Cells | DoFs | $P_1, c_f = 0.01$ | $P_1, c_f = 0.1$ | $P_1, c_f = 0.01$ | $P_1, c_f = 0.11$ | |
| 0 | 960 | 1440 | 86 | 135 | 144 | 80 | |
| 1 | 3840 | 2800 | 2013 | 923 | 410 | 536 | |
| 2 | 15360 | 17280 | 3081 | 1122 | 2027 | 1798 | |







Figure: Mesh alignment only.

Figure: Mesh alignment and *r*-adaptivity.

Figure: *r*-adaptivity only.

50 timesteps for different settings, mean values of the number of nonlinear CG iterations:

| c_{f}, p_{d} | 1, 1 | 1, 2 | 1100, 2 | cf, Pd | 1, 1 | 1, 2 | $\frac{1}{100}$, 2 | c _f , p _d | 1, 1 | 1, 2 | 1100, 2 |
|----------------|------|------|---------|--------|------|------|---------------------|---------------------------------|------|------|---------|
| Level | | | | Level | | | | Level | | | |
| 3 | 517 | 514 | 3244 | 3 | 930 | 516 | 208 | 3 | 35 | 31 | 76 |
| 4 | 623 | 637 | 2844 | 4 | 717 | 610 | 1367 | 4 | 84 | 85 | 169 |
| 5 | 1183 | 1600 | 4765 | 5 | 527 | 523 | 1218 | 5 | 255 | 248 | 591 |
| 6 | 1599 | 1754 | 4687 | 6 | 921 | 853 | 2875 | 6 | 734 | 773 | 2542 |

Jordi Paul, 03.03.2016

Computational Aspects

35/37



The Verdict

- **1** If working special purpose code is available: Use it.
- 2 If a quadratic mesh quality functional gives satisfactory results: Use it.
- **3** If a nonlinear extension of such a functional accomplishes what is needed: Use it.
- 4 If nothing else works: Have a look at nonlinear hyperelasticity.





Literature

S. Basting and M. Weismann.

A hybrid level set front tracking finite element approach for fluid-structure interaction and two-phase flow applications.

Journal of Computational Physics, 255(0):228 - 244, 2013.

P. G. Ciarlet.

Mathematical elasticity Vol. I: Three-dimensional elasticity, volume 20 of Studies in Mathematics and its Applications.

North-Holland Publishing Co., Amsterdam, 1988.

M. Grajewski, M. Köster, and S. Turek.

Numerical analysis and implementational aspects of a new multilevel grid deformation method. *Applied Numerical Mathematics*, 60(8):767–781, 2010. doi:10.1016/j.apnum.2010.03.017.



W. Huang and R. D. Russel.

Adaptive moving mesh methods, volume 174 of Applied Mathematical Sciences. Springer, 2011.



M. Rumpf.

A variational approach to optimal meshes. Numerische Mathematik, 72:523–540, 1996.

Computational Aspects





Appendix I: Variational Derivatives

Let $M : \mathbb{R} \to \mathbb{R}^{n \times n}$ be a matrix valued mapping. Recall that

$$\frac{d}{dt} \det M(t) = \det M(t) \operatorname{tr} \left(M(t)^{-1} \frac{d}{dt} M(t) \right).$$

For two variations ϕ, η using $M(t) := \nabla \phi + t \nabla \eta$ locally on each $K \in \mathcal{T}_h$, we can compute the following derivatives:

$$\frac{\partial (\|\nabla \phi\|_{F}^{2} - d)^{2}}{\partial \eta} = 4 \left(\|\nabla \phi\|_{F}^{2} - d \right) \nabla \phi : \nabla \eta$$
$$\frac{\partial \left(\det(\nabla \phi)^{p} \right)}{\partial \eta} = p \det(\nabla \phi)^{p-1} (\nabla \phi)^{-\tau} : \nabla \eta$$





Appendix II: ∇h We need to express $F = F(\nabla R_{K,n}, h(K))$ to resolve the dependency on h.

$$\begin{split} h(T) &= \sqrt[d]{\frac{c(T)}{\sum_{\kappa \in \mathcal{T}_{h}} \sum_{K \in \mathcal{T}_{h}} \det \nabla R_{K,n}(Id)}} \\ \Rightarrow \frac{\partial h(T)}{\partial x_{ij}} &= \frac{1}{d} \left(\frac{c(T)}{\sum_{\kappa \in \mathcal{T}_{h}} \sum_{K \in \mathcal{T}_{h}} \det \nabla R_{K,n}(Id)} \right)^{\frac{1}{d}-1} \left[\frac{c(T)}{\sum_{\kappa \in \mathcal{T}_{h}} \partial x_{ij}} \left(\sum_{K \in \mathcal{T}_{h}} \det \nabla R_{K,n}(Id) \right)^{\frac{1}{d}-1} \left[\frac{c(T)}{\sum_{\kappa \in \mathcal{T}_{h}} \partial x_{ij}} \left(\sum_{K \in \mathcal{T}_{h}} \det \nabla R_{K,n}(Id) \right)^{\frac{1}{d}-1} \left[\frac{c(T)}{\sum_{\kappa \in \mathcal{T}_{h}} \partial x_{ij}} \left(\sum_{K \in \mathcal{T}_{h}} \det \nabla R_{K,n}(Id) \right)^{\frac{1}{d}-1} \left[\frac{c(T)}{\sum_{\kappa \in \mathcal{T}_{h}} \partial x_{ij}} \left(\sum_{K \in \mathcal{T}_{h}} \det \nabla R_{K,n}(Id) \right)^{\frac{1}{d}-1} \left[\sum_{K \in \mathcal{T}_{h}} \nabla R_{K,n}(Id) \right] \right] \end{split}$$

where x_{ij} is the *j*th component of the local vertex *i* belonging to *T*.





Appendix III: Ugly Code

grad(0,0) = DataType(2) / DataType(3) * this->_fac_det * Math::sqrt(DataType(3)) * (x(1,1) - x(2,1)) * Math::pow(h(1), -DataType(2)) + DataType(8) / DataType(9) * this->_fac_norm * (DataType(3) * Math::pow(h(0), DataType(2)) - DataType(2) * Math::pow(x(0,0), DataType(2)) + DataType(2) * x (0,0) * x(1,0) + DataTvpe(2) * x(0,0) * x(2,0) - DataTvpe(2) * Math::pow(x(0,1), DataTvpe(2)) +DataType(2) * x(0,1) * x(1,1) + DataType(2) * x(0,1) * x(2,1) - DataType(2) * Math::pow(x(1,0), DataType(2)) + DataType(2) * x(1,0) * x(2,0) - DataType(2) * Math::pow(x(1,1), DataType(2)) + DataTvpe(2) * x(1,1) * x(2,1) - DataTvpe(2) * Math: pow(x(2,0), DataTvpe(2)) - DataTvpe(2) *Math::pow(x(2,1), DataType(2))) * Math::pow(h(0), -DataType(4)) * (-DataType(4) * x(0,0) + DataType(2) * x(1,0) + DataType(2) * x(2,0)) - DataType(2) * this->_fac_rec_det * Math::pow(DataType(2) / DataType(3) * Math::sqrt(DataType(3)) * (x(0,0) * x(1,1) - x(0,0) * x(2,1) - x (1,0) * x(0,1) + x(0,1) * x(2,0) + x(1,0) * x(2,1) - x(1,1) * x(2,0)) * Math::pow(h(1), -DataType(2)) + Math::sqrt(DataType(9) * this->_fac_reg * this->_fac_reg + DataType(12) * Math:: pow(x(0,0) * x(1,1) - x(0,0) * x(2,1) - x(1,0) * x(0,1) + x(0,1) * x(2,0) + x(1,0) * x(2,1) - x(1,0) + x(1,0)(1,1) * x(2,0), DataType(2)) * Math::pow(h(1), -DataType(4))) / DataType(3), -DataType(3)) * (DataType(2) / DataType(3) * Math::sqrt(DataType(3)) * (x(1,1) - x(2,1)) * Math::pow(h(1), -DataType(2)) + DataType(4) * Math::pow(DataType(9) * this->_fac_reg * this->_fac_reg + DataType (12) * Math:: pow(x(0,0) * x(1,1) - x(0,0) * x(2,1) - x(1,0) * x(0,1) + x(0,1) * x(2,0) + x(1,0)* x(2,1) - x(1,1) * x(2,0), DataTvpe(2)) * Math::pow(h(1), -DataTvpe(4)), -DataTvpe(1) / DataType(2)) * (x(0,0) * x(1,1) - x(0,0) * x(2,1) - x(1,0) * x(0,1) + x(0,1) * x(2,0) + x(1,0) *x(2,1) - x(1,1) * x(2,0) * Math::pow(h(1), -DataType(4)) * (x(1,1) - x(2,1)));



Appendix IV: Example n + 1: Refinement Of The Unit Circle

Boundary vertices added by refinement are moved to their appropriate place by evaluating the analytic description of the boundary, which is the unit circle.





fakultät für **m**l

Figure: The reference domain and mesh for the unit circle. Figure: Reference domain with adjusted boundary.

But what if we do not have perfect information, but just a reference mesh and a boundary parametrisation?





fakultät für **m**



Figure: Output on levels 3 and 5 using the D(u) : D(v) functional.



Figure: Output on levels 3 and 5 using hyperelasticity based functional.



Figure: Worst angle in the mesh over different levels of refinement.





Appendix V: Additional Nonlinearity

In example 3, we used c = c(K), but it really should be $c = c(\Phi^*(K))$, introducing an additional nonlinearity.







Appendix V: Cell Distribution Quality Indicator



Figure: Mesh pre and post Jordi Paul, 03.03.2016 It is $\forall K \in \mathcal{T}_h : \frac{\operatorname{vol}(K)}{\lambda(K)} = 1$ for an ideal mesh. So we compute

$$q(\Phi^*) = rac{1}{|\mathcal{T}_h|} \sum_{K \in \Phi^*(\mathcal{T}_h)} \left(1 - rac{\lambda(K)}{\mathrm{vol}(K)}
ight)$$

| Level | q(Id) | $q(\Phi^*)$ | | | |
|---------------------------------------|--------------|--------------|--|--|--|
| 4 | 7.51810E-001 | 2.90E-002 | | | |
| 5 | 7.46550E-001 | 1.71464E-002 | | | |
| 6 | 7.42708E-001 | 8.95512E-003 | | | |
| 7 | 7.41402E-001 | 4.90519E-003 | | | |
| 8 | 7.41678E-001 | 2.94606E-003 | | | |
| Table: Call size quality indicator at | | | | | |

able: Cell size quality indicator at



Appendix VI: Optimal Variations

Let $\Omega \subset \mathbb{R}^d$, d = 2, 3 and $\partial \Omega = \biguplus_{m=0}^{d-1} \partial \Omega^m$, where $\partial \Omega^0$ is a set of singular points and $\partial \Omega^m$ are relatively open, smooth *m*-dimensional manifolds.

Then Φ^* is an *optimal variation* of the partitioning \mathcal{T}_h of Ω with respect to the functional \mathcal{F} iff

$$\begin{split} \mathcal{F}(\Phi^*) &= \min_{\Phi \in V} \mathcal{F}(\Phi) \text{ where} \\ V &:= \{\Phi : \mathcal{T}_h \to \mathbb{R}^d : \Phi \in \mathcal{C}^0(\mathcal{T}_h), \forall K \in \mathcal{T}_h : \nabla \Phi_{|K} \in SL_d, \\ \forall x \in \mathcal{V}(\mathcal{T}_h) : \forall m = 0, \dots, d-1 : x \in \partial \Omega^m \Rightarrow \Phi(x) \in \partial \Omega^m \} \end{split}$$