A Two Stage CVT / Eikonal Convection Mesh Deformation Approach for Large Nodal Deformations

Stephan Schmidt[∗]

Abstract

A two step mesh deformation approach for large nodal deformations, typically arising from non-parametric shape optimization, fluid-structure interaction or computer graphics, is considered. Two major difficulties, collapsed cells and an undesirable parameterization, are overcome by considering a special form of ray tracing paired with a centroid Voronoi reparameterization. The ray direction is computed by solving an Eikonal equation. With respect to the Hadamard form of the shape derivative, both steps are within the kernel of the objective and have no negative impact on the minimizer. The paper concludes with applications in 2D and 3D fluid dynamics and automatic code generation and manages to solve these problems without any remeshing. The methodology is available as a FEniCS shape optimization add-on at `http://www.mathematik.uni-wuerzburg.de/~schmidt/femorph`.

1   Introduction

Shape optimization is a special sub-class of PDE constrained optimization. Contrary to more common problems of minimizing a functional, shape optimization naturally leads to the question of how to parameterize the unknown domain and the respective derivative or shape update. Typically, these two problems are treated separately by first defining a parameterization, usually only of the deformation but not the shape itself, and then computing the derivative with respect to the parameterization on a separate and independent mesh. Within aerospace applications, the so-called Hicks-Henne ansatz functions [12] are often used. Other examples are free-form deformation or CAD splines with their respective parameters as design unknowns.

[∗]Dr. Stephan Schmidt, University of Würzburg, Germany; stephan.schmidt@mathematik.uni-wuerzburg.de

The advantage of this approach is that the discretization of the state and adjoint equation is essentially independent of the discretization of the deformation. Furthermore, the search space is limited to the image space of the deformation, usually resulting in a guaranteed minimum regularity but also restricting the maximum design space. The downside is, however, that the mapping of a deformation parameter to the PDE solution needs to be incorporated into the differentiation process of the objective, which can be costly to compute and requires the derivative of both the parameterization as well as the derivative of the PDE solution process with respect to the input node positions. Both aspects, when taken together, are often called "mesh sensitivities". Although they can be computed efficiently [11, 16], one often needs information not available with most commercial CAD tools. Furthermore, the approach of separating the parameterization from the discretization of the PDE makes exploiting the structure of shape optimization problems difficult, as post-discretization, the problem can typically be seen as using a functional as design unknown. Other recent approaches to overcome the meshing problem are based on immersed boundaries and phase fields [10].

The idea of shape calculus [7, 27] is to directly differentiate the objective with respect to the input domain. Given sufficient regularity, typically Lipschitz-continuity of the boundary, the directional derivative stemming from this approach can be cast into the Hadamard-form, i.e. be transformed into a boundary integral expression with tangential kernel. After discretization, a descent in the objective can then be achieved by moving boundary nodes in normal direction, thereby combining deformation parameterization and PDE discretization into one. Furthermore, this approach does not require any consideration of the mesh deformation within the derivative chain, making the resulting procedure quite attractive and efficient from a numerical standpoint, as only a scalar boundary integral kernel involving primal and dual state needs to be evaluated. The downside of this approach, however, is that higher order schemes require additional considerations [24] or at least some other type of approximative Newton scheme or alternate space to find the gradient in [18], sometimes called Sobolev-gradient descent. Because tangential nodal movement is within the kernel of the Hadamard-form, the parameterization of the optimal shape of this approach is often quite uneven and heavily influenced by the node placement on the original shape, leading to the problem of redistributing surface nodes to guarantee a satisfactory discretization of the final shape. The situation is quite similar to curvature flow, for which schemes with automatic node redistribution can for example be found in [2]. The idea there is to use a modified weak form to compute the curvature flow, which automatically leads to an equidistant node spacing on curves. Well-parameterized multi-phase fluid interfaces are for example discussed in [3].

Because shape optimization usually is not intrinsic to the boundary, that is

traces of quantities defined in the volume are often required to calculate the boundary flow, and the methodology needs to extend to the three dimensional situation, we consider coupling the discretization of the continuous boundary flow with a discrete surface mesh quality measure stemming from a Voronoi partitioning usually used within mesh generation but not mesh deformation. After a well-parameterized surface is found, we use a ray-free ray tracing approach based on solving two Eikonal equations to deform the volume mesh robustly. The methodology is tested on optimizing an obstacle in an incompressible Navier–Stokes fluid, which requires very large nodal deformations. The new methodology works without re-meshing, which is highly advantageous for novel optimization methods like One-Shot [21, 25].

2 Nodal Shape Optimization and the Hadamard Theorem

2.1 Problem Definition

Shape optimization problems are a subclass of PDE-constrained optimization, where parts of the geometry, that is the domain $\Omega \subset D \subset \mathbb{R}^d$, are the design unknowns to be found. The set D is typically called the "hold-all". Within the context of nodal deformation, three typical cases can be identified:

$$\min_{\Omega} J_1(\Omega) := \int_{\Omega} f_1(x) \, dx \tag{2.1}$$

$$\min_{\Omega} J_2(\Omega) := \int_{\partial\Omega} f_2(s) \, ds \tag{2.2}$$

$$\min_{\Omega} J_3(\Omega) := \int_{\partial\Omega} \langle f_3(s), n(s) \rangle \, ds, \tag{2.3}$$

where $\partial\Omega$ is the boundary of Ω with outward facing normal n. The above notation gives rise to the notion of a "non-parametric" approach, because Ω is formally used directly as the optimization variable, whereas a parametric approach would consider $\Omega(\xi)$ to be defined by a possibly smooth parameterization with parameter ξ, typically splines. Other alternatives are treating $\Omega(\xi)$ locally explicitly as the graph of a function with parameter ξ or implicitly as a level set.

2.2 Directional Derivatives and the Hadamard Theorem

There are several approaches to define a perturbed domain Ω_t. A detailed overview, as well as an analysis of the shape differentiability and the resulting

gradient expressions can be found in [7, 27]. A very basic summary will be given here. Within the context of "perturbation of identity", a sufficiently smooth vector field $V : D \to \mathbb{R}^d$ is considered, such that a perturbed domain can be defined by

$$\Omega_\epsilon[V] := \{x + \epsilon V(x) : x \in \Omega\}. \tag{2.4}$$

Then the derivative of (2.1) - (2.3) in direction V is given by the one-sided limit

$$dJ_i(\Omega)[V] := \lim_{\epsilon \to 0^+} \frac{J_i(\Omega_\epsilon[V]) - J(\Omega)}{\epsilon} \tag{2.5}$$

and the gradient for the cases (2.1)-(2.3) above is given by

$$dJ_1(\Omega)[V] := \int_{\partial\Omega} \langle V, n \rangle \, f_1(s) \, ds \tag{2.6}$$

$$dJ_2(\Omega)[V] := \int_{\partial\Omega} \langle V, n \rangle \left(\frac{\partial f_2(s)}{\partial n(s)} + \kappa(s) f_2(s) \right) \, ds \tag{2.7}$$

$$dJ_3(\Omega)[V] := \int_{\partial\Omega} \langle V, n \rangle \, \mathrm{div} \, f_3(s) \, ds, \tag{2.8}$$

where κ denotes the mean curvature and div refers to the standard divergence operator, interpreted as the trace inside the boundary integral. The derivation of the last identity, Equation (2.8), can for example be found in [20]. Although it is also possible to work with volume formulations of the respective linearizations above, we will focus on the respective boundary expressions here, which typically requires stronger smoothness assumptions but can result in very fast numerical algorithms only operating on the boundary. Thus, one is free from including the mesh deformation chain into the derivative computation, although this is possible [11]. Furthermore, any tangential perturbation for V results in a zero directional derivative. Both properties will be exploited within the mesh reparameterization approach discussed here, such that the repair step is always inside the kernel of the objective.

One achieves descent in the objective J_i by iterating over sets in accordance to i.e. a steepest descent algorithm by updating

$$\partial\Omega_{k+1} := \{s + g_i(s)n(s) : s \in \partial\Omega_k\}, \tag{2.9}$$

where g_i denotes the respective scalar integration kernel from (2.6)-(2.8) above. If Ω is a tessellated domain, i.e. a finite element or finite volume mesh, then the above formula provides an algorithm of how to move the surface vertices into the vertex normal direction to achieve descent. However, the resulting point

distribution based on the above algorithm is typically rather uneven and irregular. Also, there is no obvious strategy of how to deform the volume mesh and standard approaches quickly lead to compressed or collapsed cells due to the typically very local movement of the boundary. As part of this work, a two stage centroidal voronoi surface reparameterization coupled to an Eikonal convection mesh deformation will be studied.

3 From Remeshing to CVT / Eikonal Mesh Deformation

The two typical problems with mesh based nodal shape optimization are shown in Figure 3.1: Using an update based on equation (2.9) naively results in a progressive deterioration of the parameterization during the optimization morph. Furthermore, a naive component-wise Laplacian mesh deformation does not prevent the occurrence of collapsed cells. Each of these two problems will be addressed separately.

3.1 Centroidal Voronoi Reparameterization

Since the main focus of this work is to achieve a robust mesh deformation for nodal shape optimization, we assume that an initial triangulated surface Γ of the object to be optimized and a tessellation of the surrounding polygonal domain Ω is already given, usually by some form of Delaunay discretization. The dual of the Delaunay triangulation is the centroidal Voronoi tessellation. This gives rise to define the badness of a mesh consisting of k vertices $\{x_i : i = 1, ..., k\}$ by

$$\mathcal{F}(\{x_i\}, W_i) := \sum_{i=1}^{k} \int_{W_i} \rho(x)\|x - x_i\|_2^2 \, dx, \qquad (3.1)$$

where $\Omega = \cup_{i=1}^{k} W_i$ is a discretization of Ω into k disjunct subsets. The set $\{x_i\}$ is also sometimes called the set of generators. Finally, ρ is a varying density function to prescribe areas where a local clustering of points is desired. Typically, the sets W_i are the dual cells of the triangulated primal mesh with vertices x_i, as shown in Figure 3.2.

It can be shown [8] that a necessary condition for $\mathcal{F}$ to be minimized is that the dual patches W_i are the Voronoi regions W_i^*

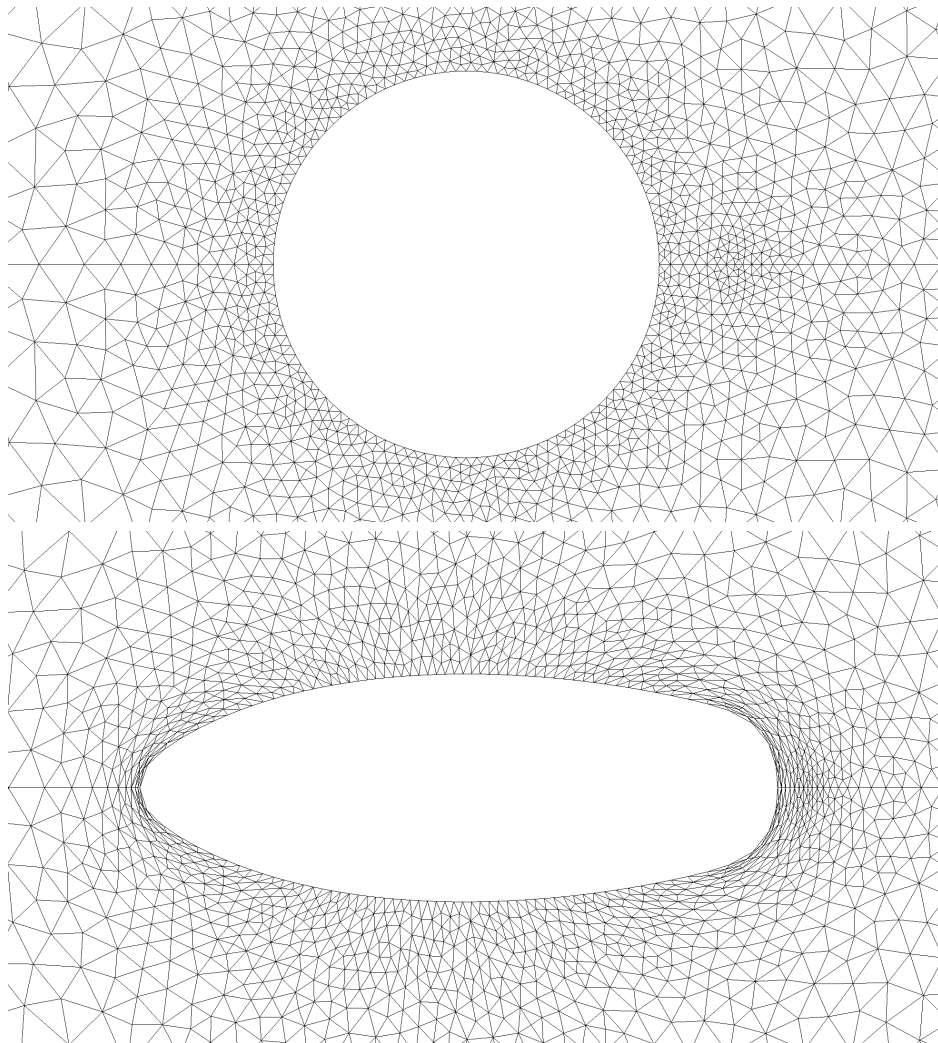$$W_i^* := \{x \in \Omega : \|x - x_i\| < \|x - x_j\|, j = 1, ..., k, i \neq j\}$$

Fig. 3.1: The initial mesh and the best optimum achievable when using a naive descent based on equation (2.9). Note the very coarse surface parameterization at the front and rear as well as the collapsed cells in the volume.

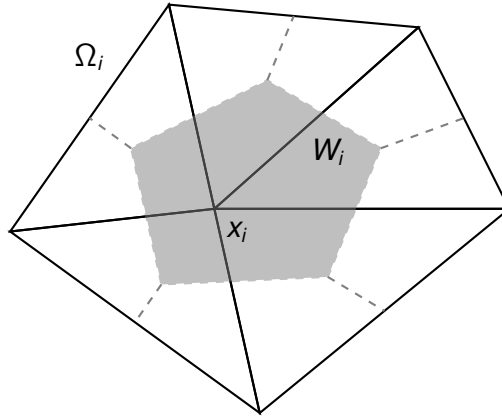corresponding to x_i. Simultaneously, the vertices x_i are the centroids of the cor-

Fig. 3.2: Triangle patch Ω_i of the mesh in solid black lines. Dual cell W_i based on triangle centroids around generator x_i in gray. Primal mesh not Delaunay, because x_i is not the centroid of the dual patch.

responding W_i, because

$$\nabla_{x_i}\mathcal{F}(\{x_i\}, W_i) = 2 \int_{W_i} \rho(x) \, dx \cdot x_i - 2 \int_{W_i} \rho(x)x \, dx \overset{!}{=} 0$$

$$\Rightarrow x_i = \frac{\int_{W_i} \rho(x)x \, dx}{\int_{W_i} \rho(x) \, dx},$$

meaning choosing the centroid of W_i for x_i is a critical point if W_i is considered fixed.

Of equal importance to the node spacing is the node connectivity of the primal mesh. Vertices x_i and x_j of the primal mesh are connected by an edge if and only if the dual regions they generate are adjacent. Thus, dual patches W_i in (3.1) correspond to edges of the primal mesh. Because the goal here is to ensure a high quality mesh deformation during nodal shape optimization, a change in mesh connectivity is highly unwanted, corresponding to a remeshing procedure rather than a mesh deformation approach. Thus, we follow the notion of [5] by considering only the vertex positions, but not their connectivity, as the defining factor of mesh badness during nodal shape optimization. Contrary to the considerations before, patches Ω_i of the primal triangular mesh, rather than the Voronoi patches of the dual mesh, are used to define an alternate mesh badness measure. This is of considerable numerical advantage, because a costly computation of the Voronoi tessellation is omitted. The resulting mesh badness

measure is thus given by

$$\mathcal{F}(\{x_i\}) := \sum_{i=1}^{k} \int_{\Omega_i} \rho(x)\|x - x_i\|_2^2 \, \mathrm{d}x \tag{3.2}$$

and successively setting the mesh vertices to the critical points of (3.2) results in the CVT-I smoother as discussed in [5].

3.2 Surface Mesh Badness for Nodal Shape Optimization

For the nodal shape optimization problem under consideration here, the above badness measure needs to be adapted, such that a vertex distribution that is a critical point of the badness measure is inside the kernel of the shape derivative. Consequently, the mesh repair step is a tangential movement of surface mesh nodes only and does not interfere with optimality, because mesh quality step and shape optimality/feasibility step are orthogonal to each other, due the shape derivatives (2.6)–(2.8) being invariant under tangential deformation directions V.

A natural process when adapting (3.2) to a surface mesh would be to change the $\|.\|_2$-norm to the geodetic distance between points, as using the Euclidian distance will not result in the critical vertex set $\{x_i\}$ to be on the surface $\Gamma := \partial\Omega$. The computation of geodetic distances and their derivatives can be quite costly. However, the Euclidian distance within the local tangent plane to $\partial\Omega_i$ at x_i, that is $T_{x_i}\partial\Omega_i$, also results in critical points of (3.2) to stay within said tangent plane, as will be discussed next.

For any point x in $\Omega_i \subset \mathbb{R}^d$, the projection to the tangent plane $T_{x_i}\Gamma$ is given by

$$x_\Gamma := x - \langle x, n(x_i)\rangle \cdot n(x_i),$$

where $n(x_i)$ is the normal at x_i. Using this definition, the surface mesh badness considered here is given by

$$\mathcal{F}(\{x_i\}) := \sum_{i=1}^{k_\Gamma} \int_{\Gamma_i} \rho(s)\|(s - x_i)_\Gamma\|_2^2 \, \mathrm{d}s \tag{3.3}$$

and the critical set is given by points x_i^* such that

$$\int_{\Gamma_i} \rho(s) \left(-(s - x_i^*)_\Gamma - \langle s - x_i^*, n\rangle n'[x_i](s - x_i^*)_\Gamma \right) \, \mathrm{d}s \stackrel{!}{=} 0, \tag{3.4}$$

where

$$n'[x_i] := \nabla_x n(x) = \left(\frac{\partial}{\partial x} n(x_i) \right)^T \in \mathbb{R}^{d \times d}$$

denotes the variation of the normal with respect to a shift of the origin point x_i of the tangent plane. It can be shown [7, 20, 27] that this object also lies within the tangent plane. The above relationship directly follows from

$$\nabla_{x_i} \|(s - x_i)_\Gamma\|_2^2$$

$$= \nabla_{x_i} \sum_{\ell=1}^{d} \left(s^\ell - x_i^\ell - \langle s - x_i, n \rangle n^\ell \right)^2$$

$$= \sum_{\ell=1}^{d} 2 \left(s^\ell - x_i^\ell - \langle s - x_i, n \rangle n^\ell \right) \nabla_{x_i} \left[s^\ell - x_i^\ell - \langle s - x_i, n \rangle n^\ell \right]$$

$$= -2(s - x_i)_\Gamma - 2 \left\langle (s - x_i)_\Gamma, \nabla_{x_i} \left[\langle s - x_i, n \rangle \cdot n \right] \right\rangle,$$

where upper indices denote vector components. Furthermore,

$$\left\langle (s - x_i)_\Gamma, \nabla_{x_i} \left[\langle s - x_i, n \rangle \cdot n \right] \right\rangle$$

$$= - \langle (s - x_i)_\Gamma, n \rangle \cdot n + \langle (s - x_i)_\Gamma, n \rangle n'[x_i] \cdot (s - x_i) + \langle s - x_i, n \rangle n'[x_i] \cdot (s - x_i)_\Gamma$$

$$= \langle s - x_i, n \rangle n'[x_i] \cdot (s - x_i)_\Gamma,$$

where the relationships $(a - b)_\Gamma = a_\Gamma - b_\Gamma$ and $\langle a_\Gamma, n \rangle = 0$ for some vectors a and b have been used.

With the same reasoning used when assuming the patches to be independent of the generator in making the transition from (3.1) to (3.2), we also suppose the tangential plane, and thus the normal to it, to be independent of the generator in each repair iteration, which greatly simplifies Equation (3.4) to

$$x_{i,\Gamma}^* = \frac{\int\limits_{\Gamma_i} \rho(s) s_\Gamma \, ds}{\int\limits_{\Gamma_i} \rho(s) \, ds}, \tag{3.5}$$

which will be used as our tangential mesh repair surface node reparameterization iteration. The resulting improved surface node spacing can be seen in Figure 3.3. Although the surface mesh point distribution has improved greatly, the issue of collapsed cells still has to be addressed.

3.3 Eikonal Convection Mesh Deformation

In principle, the above methodology can also be used to deform the volume mesh mimicking the node spacing aspect of the Delaunay meshing procedure. How-
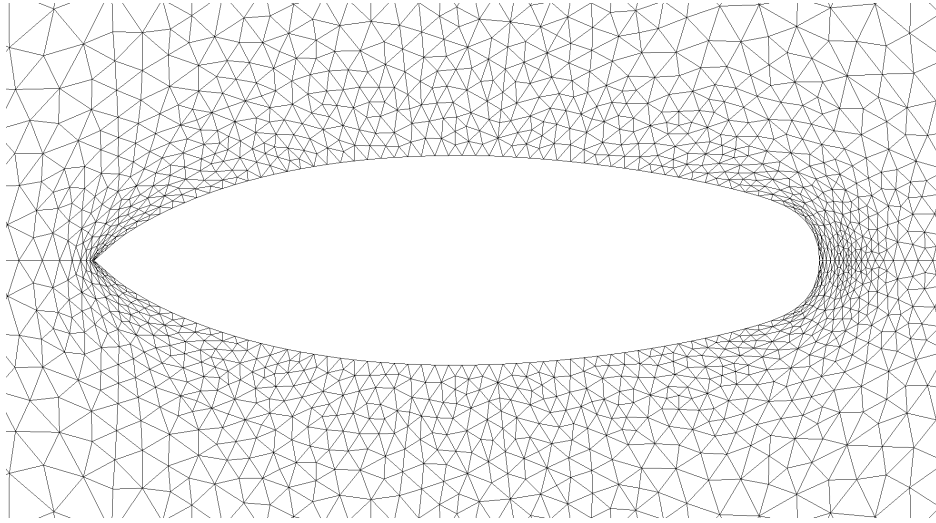
Fig. 3.3: Optimal Shape of the Navier–Stokes problem with surface reparameter-
ization. The volume mesh still has collapsed cells.

ever, since an even spacing of volume mesh nodes is seldom desired, this ap-
proach would first require a reconstruction of the original density ρ, which deter-
mined the initial node spacing of the original mesh. The most common approach
to adapt the volume mesh to a new boundary is to solve a Laplace-like equation,
in particular the component wise Laplacian

$$-\epsilon \Delta v = 0 \qquad \text{in } \Omega$$
$$v = g \cdot n \quad \text{on } \partial\Omega,$$

where $g \cdot n$ is the desired boundary deformation from (2.9). This approach, how-
ever, often results in collapsed cells and distorted elements if the deformation
is too large. Numerous applications within fluid-structure interaction problems
have demonstrated that linear elasticity or certain harmonic operators provide
an excellent approach to deform the volume mesh [6, 9, 29]. Common to those
problems is, however, a fairly global and smooth boundary deformation and often
those approaches based on elliptic or parabolic PDEs have problems to preserve
cell volumes under the highly local deformations occurring in nodal shape opti-
mization. Finally, a very interesting recent approach of volume mesh deformation
is to base the PDE to be solved for calculating the nodal displacement vectors on
a spectrally equivalent PDE that approximates the pseudo-differential operator
nature of the design to state mapping of the optimization problem [1, 22]. That
way, extending the deformation to the volume and calculating an approximation
to the Newton descent direction happens simultaneously [26].

As an alternate approach, volume mesh deformation based on a convection-diffusion equation

$$-\epsilon_1 \Delta v + \text{div}\,(v \cdot b) = 0 \qquad \text{in } \Omega \tag{3.6}$$
$$v = g \cdot n \quad \text{on } \partial\Omega$$

is considered here. This idea is to use the convective wind $b := \nabla \epsilon_2$ to project the deformation displacement orthogonal to the surface tangents into the volume mesh. Thus, we follow an inverse ray tracing approach by not calculating an orthogonal projection of every volume node to the design surface, but rather follow an infinite number of orthogonal rays from the design surface into the volume. A convenient way of achieving this is to directly solve the Eikonal equation without rays [17]:

$$h\Delta\epsilon + \left\| \nabla\epsilon \right\|_2^2 = 1 \quad \text{in } \Omega, \tag{3.7}$$

where h is a small stabilization parameter. When specifying a Dirichlet zero boundary values for ϵ, the numerical value of ϵ is a non-dimensional travel time to the boundary when locally following the shortest path ray given by $\nabla\epsilon$, which is thus an approximation to the orthogonal projection. Alternatively, the value $\nabla\epsilon$ can be interpreted as a reasonable extension of the surface normal into the volume, which is often required in nodal shape optimization.

In order to compute the actual mesh deformation in accordance with (3.6), Equation (3.7) is solved twice, once for ϵ_1 by setting Dirichlet zero boundary conditions on the far-field and once for ϵ_2 by setting Dirichlet zero boundary conditions on the unknown obstacle. Natural boundary conditions are used on the respective other boundaries. Thus, $\epsilon_1(x)$ indicates the travel time, i.e. the distance, to the far-field, while $\epsilon_2(x)$ will indicate the distance to the unknown shape to be optimized. The actual displacement vectors are then found by solving

$$-\alpha\epsilon_1^2 \Delta v + \text{div}\,(\beta v \cdot \nabla\epsilon_2) = 0 \qquad \text{in } \Omega \tag{3.8}$$
$$v = g \cdot n \quad \text{on } \Gamma$$
$$v = 0 \qquad \text{on } \partial\Omega \setminus \Gamma$$

where α and β are two real parameters to be chosen based on the actual geometry. Diffusivity is scaled down with the square of the distance to the shape changing obstacle, making the convective term dominant in close vicinity. The above equation is solved using 2nd order SUPG-Finite elements. The non-linearity in the Eikonal equation is solved via Newton's method. The resulting mesh is shown in Figure 3.4 and is very well behaved.
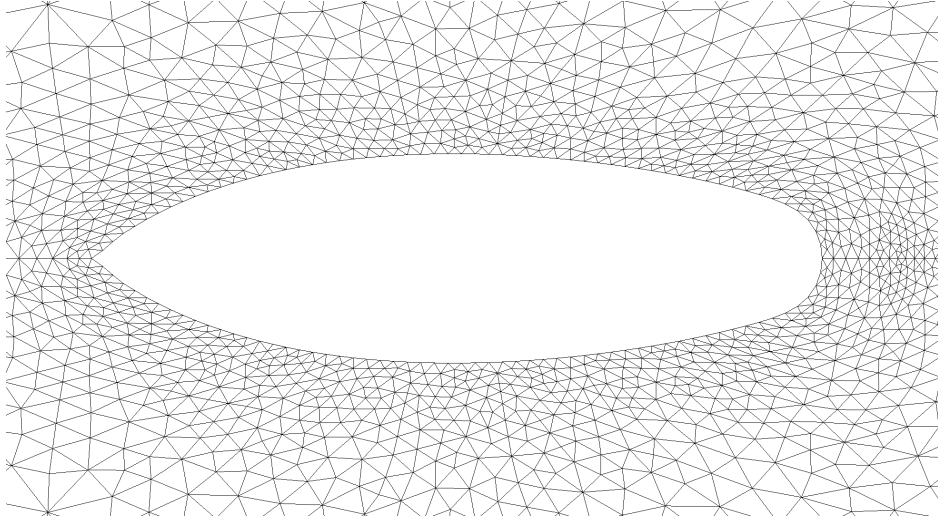
Fig. 3.4: The final mesh using both the CVT surface repair and the Eikonal mesh deformation. The surface is nicely parameterized and all volume cells are well shaped. No remeshing is necessary.

4 Application Example: Energy Minimization in a Navier-Stokes Fluid

As an application example, shape optimization in a Navier–Stokes fluid is considered. The problem is well-understood [4, 14, 22, 23] and could be seen as the extension of the DFG simulation benchmark [19] into shape optimization. Because well-spaced surface node positions are much more difficult so achieve for surfaces Γ within a 3D domain Ω, the following problem will be considered in both two and three dimensions.

$$\min_{(u,p,\Omega)} J(u, p, \Omega) := \int_{\Omega} \mu \sum_{i,j=1}^{3} \left(\frac{\partial u_i}{\partial x_j} \right)^2 \, dA \tag{4.1}$$

subject to

$$-\mu \Delta u + \rho u \nabla u + \nabla p = 0 \quad \text{in} \quad \Omega$$
$$\text{div } u = 0$$
$$u = u_+ \quad \text{on} \quad \Gamma_+$$
$$u = 0 \quad \text{on} \quad \Gamma_0$$
$$pn - \mu \frac{\partial u}{\partial n} = 0 \quad \text{on} \quad \Gamma_-$$
$$\text{vol} = V_0$$
$$\text{geo}(\Omega) = 0, \tag{4.2}$$

where vol denotes the volume and geo summarizes additional geometric constraints such as fixing the centroid of Ω and preserving the symmetry planes. Especially the latter is quite important, as otherwise an incorrect discrete normal at sharp edges [13] and general finite element approximation errors [15] tend to result in a problematic behavior at spurious or physical geometric singularities. Not considering the volume and geometric constraints, the shape derivative of the above problem is given by

$$dJ(u, p, \Omega) = \int_{\Gamma_0} \langle V, n \rangle \left[-\mu \sum_{i=1}^{3} \frac{\partial \lambda_i}{\partial n} \frac{\partial u_i}{\partial n} + \left(\frac{\partial u_i}{\partial n} \right)^2 \right] dS, \tag{4.3}$$

where (λ, λ_p) solves the adjoint Navier–Stokes equations

$$-\mu \Delta \lambda_i - \rho \sum_{j=1}^{3} \left(\frac{\partial \lambda_j}{\partial x_i} u_j + \frac{\partial \lambda_i}{\partial x_j} u_j \right) - \frac{\partial \lambda_p}{\partial x_i} = -2\mu \Delta u_i \quad \text{in} \quad \Omega$$
$$\text{div } \lambda = 0$$
$$\lambda = 0 \quad \text{on} \quad \Gamma_+$$
$$\lambda = 0 \quad \text{on} \quad \Gamma_0$$
$$\mu \frac{\partial \lambda_i}{\partial n} + \rho \left(\sum_{j=1}^{3} \lambda_j u_j n_i + \lambda_i u_j n_j \right) + \lambda_p n_i = 0 \quad \text{on} \quad \Gamma_-.$$

Both primal and dual Naiver–Stokes equations are implemented in Python FEniCS. The non-linearity of the primal problem is either solved using a Newton-Iteration or by successively solving Oseen problems stemming from a Picard-linearization. From a linear-algebra perspective, the resulting saddle-point problem is either directly factorized using the parallel sparse direct solver "mumps" or solved iteratively using a continuous pressure Schur-complement iteration inside an implicit Euler time stepping to steady state [28]. Initial and optimal shape with and without the Eikonal mesh deformation and repair step are shown in Figure 4.1.
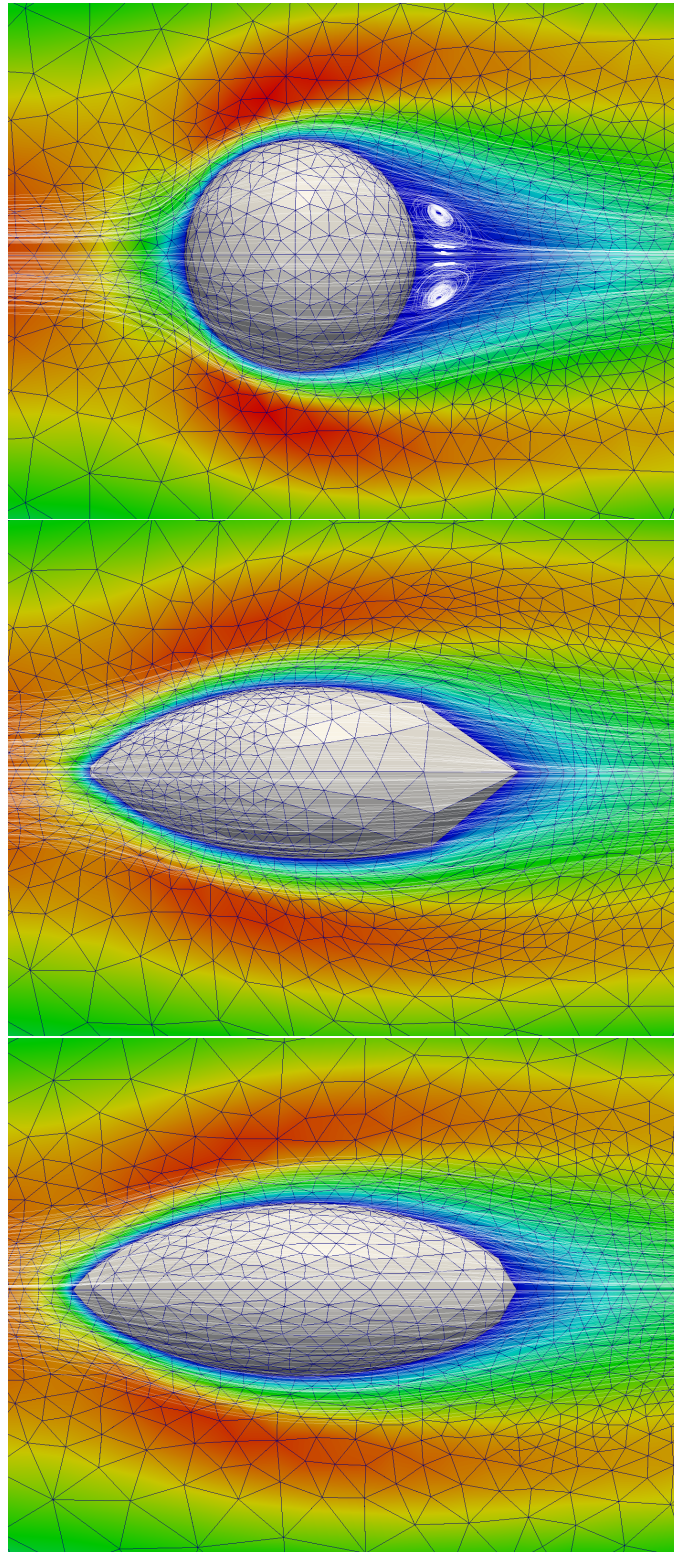
Fig. 4.1: Top: Initial 3D geometry. Middle: Naive deformation, no repair, Laplace deformation. Bottom: Full CVT/Eikonal deformation and repair. No remeshing is necessary.

In 2D, the domain is a rectangle with a half-axis of 1.5 around a half-circle of radius 0.5 at the coordinate origin. The three dimensional geometry is created by rotating this setup around the x-axis by 90 degrees. This half-domain, or quarter-domain in 3D, is discretized using an unstructured mesh. The actual computational mesh is afterwards created by mirroring these half-meshes, thereby creating a perfectly symmetric mesh that can still be used to represent non-symmetric solutions, e.g. separated flows and vortex streets. The primal and adjoint Navier–Stokes equation is discretized using LBB-stable P2P1-Taylor Hood elements, leading to $131,790$ unknowns for the primal and dual velocities each and $5,728$ unknowns for each primal and dual pressure in 3D. For the two dimensional test-cases we arrive at $17,324$ and $2,221$ unknowns respectively. The Eikonal equation and the convection-diffusion equation appearing during the mesh adaptation step are also solved using 2nd order SUPG-stabilized finite elements. All PDEs are discretized using the Python implementation of FEniCS, which also automatically generates the Newton solver for the Eikonal equation.

The reduced Hessian of the problem is approximated by a Laplace-Beltrami operator, such that a smoothed gradient descent scheme is employed. For more details see [22]. Volume and miscellaneous geometric constraints are enforced using a simple projection during each shape update. The actual algorithm is described in Algorithm 1 in all detail.

It is worth noting that, contrary to the aerodynamic force minimization using boundary integrals, the energy objective, Equation (4.1), scales with the domain size Ω, which makes an absolute comparison of objective difficult. Nevertheless, for the results presented here, the objective is reduced from 266.1282 to 244.9740 in the 3D case, and from 195.6071 to 131.3547 in the 2D situation, a relative decrease comparable to [22]. The initial and respective final shapes are shown in Figure 4.1. With the CVT tangential movement and Eikonal mesh deformation, the optimal shape is nicely parameterized and the whole optimization works successfully, maintaining well-behaved volume cells without re-meshing.

5 Conclusions

A two level mesh repair and mesh deformation strategy for large nodal shape optimization is considered and exemplified using the energy minimization in an incompressible Navier–Stokes fluid. A well-parameterized surface is maintained by introducing a mesh quality constraint into the nodal boundary update based on minimizing the tangential projection of the mesh badness criterion stemming from the centroidal Voronoi tessellation, the mesh dual of the Delaunay triangulation. The methodology is useable as a surface mesh deformation scheme, rather than a re-meshing approach, because the patch definition is not changed within

$q_k \leftarrow$ solution of state equation in current domain Ω_k
$\lambda_k \leftarrow$ solution of adjoint equation in current domain Ω_k
Calculate boundary displacement field $d_1 \leftarrow g(q_k, \lambda_k)$ as in Equation (2.9)
Calculate approximate Newton update: $d_2 \leftarrow$ solution of $(\delta \Delta_\Gamma + I)d_2 = -d_1$
Project d_2 to geometric constraints geo and volume constraint vol
Calculate intermediate next boundary $\partial \Omega_{k+\frac{1}{2}} = \{x + d_2(x) \cdot n(x) : x \in \partial \Omega_k\}$
loop
    Calculate intermediate surface node positions $x^*_{i,\Gamma}$ by solving (3.5).
    Calculate tangential offset and badness $\tau(x) \leftarrow x^*_{i,\Gamma} - (x + d_2(x) \cdot n(x))$
    if $\|\tau\| \leq \epsilon$ **then**
      break
    end if
    $\partial \Omega_{k+\frac{1}{2}} \leftarrow \{x + d_2(x) \cdot n(x) + \tau : x \in \partial \Omega_k\}$
end loop
Define final normal and tangential boundary displacement $d_3 := d_2 \cdot n + \tau$
$\epsilon_1 \leftarrow$ solution of Eikonal (3.7) in Ω_k with Dirichlet zero on far-field
$\epsilon_2 \leftarrow$ solution of Eikonal (3.7) in Ω_k with Dirichlet zero on design
Define $b := \nabla \epsilon_2$ ray tracing directions in Ω_k
Calculate full deformation in Ω, i.e. $d_4 \leftarrow$ solution of (3.6)
Update domain $\Omega_{k+1} = \{x + d_4(x) : x \in \Omega_k\}$
if Ω_{k+1} fulfills convergence **then**
    exit
else
    loop
end if

Algorithm 1: Nodal shape optimization scheme with CVT step and Eikonal deformation.

each quality preservation step, meaning the cell connectivity is not changed. Furthermore, the tangential repair step is in the kernel of the shape derivative, thus a reparameterization has no impact on optimality in a continuous interpretation.

In a second step, the boundary deformation is propagated into the volume mesh by solving a convection–diffusion equation with wind direction based on a ray tracing approach, which is achieved by solving an Eikonal equation. The methodology is exemplified by minimizing the energy dissipation in a Navier–Stokes fluid completely without remeshing in both two and three dimensions.

References

[1] E. Arian and S. Ta'asan. Analysis of the Hessian for aerodynamic optimization: Inviscid flow. Technical Report 96-28, Institute for Computer Applications in Science and Engineering (ICASE), 1996.

[2] J. W. Barrett, H. Garcke, and R. Nürnberg. The approximation of planar curve evolutions by stable fully implicit finite element schemes that equidistribute. *Numerical Methods for Partial Differential Equations*, 27(1):1–30, 2011.

[3] J. W. Barrett, H. Garcke, and R. Nürnberg. A stable parametric finite element discretization of two-phase navier–stokes flow. *Journal of Scientific Computing*, 2014.

[4] C. Brandenburg, F. Lindemann, M. Ulbrich, and S. Ulbrich. A continuous adjoint approach to shape optimization for navier stokes flow. In *Optimal Control of Coupled Systems of Partial Differential Equations*, volume 158 of *International Series of Numerical Mathematics*, pages 35–56. Birkhäuser Basel, 2009.

[5] L. Chen. Mesh smoothing schemes based on optimal delaunay triangulations. In *13th International Meshing Roundtable*, pages 109–120, Williamsburg, VA, 2004. Sandia National Laboratories.

[6] P. G. A. Cizmas and J. I. Gargoloff. Mesh generation and deformation algorithm for aeroelasticity simulations. *Journal of Aircraft*, 45(3):1062–1066, 2008.

[7] M. C. Delfour and J.-P. Zolésio. *Shapes and Geometries: Analysis, Differential Calculus, and Optimization*. Advances in Design and Control. SIAM Philadelphia, 2001.

[8] Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.

[9] R. P. Dwight. Robust mesh deformation using the linear elasticity equations. In *Proceedings of the Fourth International Conference on Computational Fluid Dynamics (ICCFD)*, pages 401–406, Ghent, Belgium, 10-14 July 2006. Springer Berlin Heidelberg.

[10] H. Garcke, C. Hecht, and M. Hinze. Numerical approximation of phase field based shape and topology optimization for fluids. Technical report, Preprint 08/2014 University Regensburg, 2014.

[11] N. Gauger, A. Walther, C. Moldenhauer, and M. Widhalm. Automatic differentiation of an entire design chain for aerodynamic shape optimization. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, 96:454–461, 2007.

[12] R. M. Hicks and P. A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15(7):407–412, 1978.

[13] C. Lozano. Discrete surprises in the computation of sensitivites from boundary integrals in the continuous adjoint approach to inviscid aerodynamic shape optimization. *Computers & Fluids*, 56, 2012.

[14] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Numerical Mathematics and Scientific Computation. Clarendon Press Oxford, 2001.

[15] P. Morin, R. H. Nochetto, M. S. Pauletti, and M. Verani. Adaptive finite element method for shape optimization. *EASIM: Control, Optimisation and Calculus of Variations*, 18:1122–1149, 2012.

[16] E. J. Nielsen and M. A. Park. Using an adjoint approach to eliminate mesh sensitivities in computational design. *AIAA Journal*, 44(5):948–953, 2006.

[17] R. L. Nowack. Wavefronts and solutions of the eikonal equation. *Geophysical Journal International*, 110:55–62, 1992.

[18] B. Protas. Adjoint-based optimization of pde systems with alternative gradients. *Journal of Computational Physics*, 227(13):6490–6510, 2008.

[19] M. Schäfer and S. Turek. Benchmark computations of laminar flow around a cylinder. In E. Hirschel, editor, *Flow Simulation with High-Performance Computers II. DFG priority research program results 1993–1995*, number 52 in Notes on Numerical Fluid Mechanics, pages 547–566. Vieweg, 1996.

[20] S. Schmidt. *Efficient Large Scale Aerodynamic Design Based on Shape Calculus*. PhD thesis, University of Trier, Germany, 2010.

[21] S. Schmidt, C. Ilic, N. Gauger, and V. Schulz. Airfoil design for compressible inviscid flow based on shape calculus. *Optimization and Engineering*, 12(3):349–369, 2011.

[22] S. Schmidt and V. Schulz. Impulse response approximations of discrete shape Hessians with application in CFD. *SIAM Journal on Control and Optimization*, 48(4):2562–2580, 2009.

[23] S. Schmidt and V. Schulz. Shape derivatives for general objective functions and the incompressible Navier-Stokes equations. *Control and Cybernetics*, 39(3):677–713, 2010.

[24] V. Schulz. A riemannian view on shape optimization. *Foundations of Computational Mathematics*, 14:483–501, 2014.

[25] V. Schulz and I. Gherman. One-shot methods for aerodynamic shape optimization. In N. Kroll, D. Schwamborn, K. Becker, H. Rieger, and F. Thiele, editors, *MEGADESIGN and MegaOpt — German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design*, volume 107 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 207–220. Springer, 2009.

[26] M. Siebenborn. *Discontinuous Galerkin Approaches for HPC Flow Simulations on Stream Processors*. PhD thesis, University of Trier, Germany, 2013.

[27] J. Sokolowski and J.-P. Zolésio. *Introduction to Shape Optimization: Shape Sensitivity Analysis*. Springer Berlin Heidelberg, 1992.

[28] S. Turek. *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*. Springer Berlin, 1999.

[29] T. Wick. Fluid-structure interactions using different mesh motion techniques. *Computers and Structures*, 89:1456–1467, 2011.