



Fluid dynamic shape optimization using self-adapting nonlinear extension operators with multigrid preconditioners

Jose Pinzon¹ · Martin Siebenborn¹

Received: 18 August 2021 / Revised: 15 March 2022 / Accepted: 23 March 2022 /

Published online: 9 April 2022

© The Author(s) 2022

Abstract

In this article we propose a scalable shape optimization algorithm which is tailored for large scale problems and geometries represented by hierarchically refined meshes. Weak scalability and grid independent convergence is achieved via a combination of multigrid schemes for the simulation of the PDEs and quasi Newton methods on the optimization side. For this purpose a self-adapting, nonlinear extension operator is proposed within the framework of the method of mappings. This operator is demonstrated to identify critical regions in the reference configuration where geometric singularities have to arise or vanish. Thereby the set of admissible transformations is adapted to the underlying shape optimization situation. The performance of the proposed method is demonstrated for the example of drag minimization of an obstacle within a stationary, incompressible Navier–Stokes flow.

Keywords Aerodynamic shape optimization · Method of mappings · Geometric multigrid · Parallel computing

Mathematics Subject Classification 35Q93 · 49Q10 · 65M55 · 65Y05 · 65K10

1 Introduction

PDE-constrained shape optimization is a mathematical tool to obtain an optimal contour for a randomly-shaped obstacle subject to physical phenomena described by a partial differential equation (PDE). This is achieved by the evaluation of sensitivities of a shape functional $j(y, \Omega)$, which depends on the state variable y and

✉ Jose Pinzon
jose.pinzon@uni-hamburg.de

Martin Siebenborn
martin.siebenborn@uni-hamburg.de

¹ University Hamburg, Bundesstr. 55, 20146 Hamburg, Germany

the domain Ω . The set of admissible shapes, G_{adm} , will be implicitly defined within the method of mappings by the set of admissible transformations. The functional is constrained by one or several PDEs, among them a state equation $E(y, \Omega) = 0$, which is fulfilled by the state variable y . In this article, we focus on the incompressible flow case, which is described by Navier–Stokes equations. Here the objective is to minimize the energy dissipation around an obstacle in terms of its shape and additional geometrical constraints. Building on a long history ranging back several decades (see for instance Jameson 2003; Giles and Pierce 2000; Mohammadi and Pironneau 2010), the field of shape optimization with fluid dynamics applications is still very active today following various approaches (e.g. Schmidt et al. 2013; Müller et al. 2021; Garcke et al. 2016; Fischer et al. 2017). For an overview of shape optimization constrained by PDEs we refer the reader to Sokolowski and Zolesio (2012), Allaire et al. (2021), Delfour and Zolésio (2001). The iterative application of deformation fields to a finite element mesh can lead to distortions and loss of mesh quality, as studied by Dokken et al. (2019), Etling et al. (2018), Iglesias et al. (2018), Blauth (2021). This becomes particularly disruptive for numerical algorithms if there are large deformations leading from the reference domain to the optimal configuration. Several approaches, especially in recent studies, have been proposed to prevent this. For instance, the use of penalized deformation gradients in interface-separated domains helps maintain mesh quality but might still lead to element overlaps when taken to the limit Siebenborn and Vogel (2021). Other approaches rely on remeshing the domain, as for instance in Wilke et al. (2005). More recent efforts on this area make use of pre-shape calculus to allow for the simultaneous optimization of both the shape and mesh quality of the grid Luft and Schulz (2021a).

Although the variable of the optimization problem is only the contour of the shape, the surrounding space plays a crucial role since it describes the domain for the physical effects. Due to the Hadamard–Zolésio structure theorem (see for instance Sokolowski and Zolesio 2012) changes of the objective function under consideration can be traced back to deformations of the shape, which are orthogonal to its boundary. This has been recognized as a source for decreasing mesh quality and is addressed by many authors, for instance by also allowing displacements tangent to the shape surface (cf. Luft and Schulz 2021b). In contrast to the statement of the structure theorem, from a computational point of view, it can be favorable to extend surface deformations into the entire surrounding domain instead of building a new discretization around the improved shape after each descent step, i.e. avoid remeshing on each new iteration. In recent works it has become popular to reuse the domain around the shape, which describes the domain of the PDE, for the representation of Sobolev gradients (e.g. Schulz et al. 2016). Typically, elliptic equations are solved in this domain in order to represent the shape sensitivity as a gradient with respect to a Hilbert space defined therein Dokken et al. (2020), Gangl et al. (2015). The benefit of this approach is that the resulting deformation field not only serves as a deformation to the obstacle boundary, but can also be utilized as a domain deformation. Thus, the discretization for the next optimization step is obtained without additional computational cost.

At this point, two different approaches can be distinguished within the context of the finite element method. On the one hand, the computed gradient can be used

directly as a deformation to the domain after each optimization step which can be seen as changing the reference domain iteratively. On the other, in the so called method of mappings Murat and Simon (1976), the reference domain is fixed and the shape optimization problem is interpreted via the theory of optimal control. These are implemented through the definition of a variable around the surface of the shape to be optimized and its connection with the deformation field affecting the whole domain through a so called extension operator. The solution of which results in the optimal deformation field for both the target shape and its surrounding domain. For an application of this method to aerodynamic shape optimization see for instance Fischer et al. (2017).

We can oppose these two approaches as

$$\begin{array}{ll} \min_{\Omega \in G_{\text{adm}}} j(y, \Omega) & \min_{F \in F_{\text{adm}}} j[y, F(\Omega)] \\ \text{s.t. } e(y, \Omega) = 0 & \text{s.t. } e[y, F(\Omega)] = 0 \end{array} \quad (1)$$

where either a set of admissible domains G_{adm} or a set of admissible transformations F_{adm} has to be defined. A link between these two can then be established via

$$G_{\text{adm}} := \{F(\Omega) : F \in F_{\text{adm}}\} \quad (2)$$

in terms of a given reference configuration Ω .

The approach we propose here is based on the research carried out in Haubner et al. (2021). Compared to iteratively updating the shape, it offers the possibility to require properties of the deformation from reference to optimal shape. Moreover, it reformulates the optimization over a set of admissible transformations $F \in F_{\text{adm}}$, which enables us to carry out the optimization procedure on the reference domain. Additionally, it has been documented that such extension operators are possible without the need for heuristic parameter tuning. In Onyshkevych and Siebenborn (2021) an additional nonlinear term is introduced to the elliptic extension equation allowing for large deformations while preserving mesh quality, and preventing element self-intersections and degeneration. In shape optimization this occurs when trying to obtain an optimal shape for an obstacle surrounded by media, for which the creation or removal of a singularity on the obstacle's boundary is necessary.

In the present work we focus on applying parallel solvers for the solution of PDEs in large distributed-memory systems. This stems from the fact that the discretizations will lead to a very large number of degrees of freedom (DoFs), for which the application of the geometric multigrid method (see for instance Hackbusch 1985) guarantees mesh-independent convergence on the simulations. Its application within the context of parallel computing towards the solution of PDEs is an area of ongoing research Reiter et al. (2013), Gmeiner et al. (2014), Baker et al. (2011). The feature of mesh-independent convergence is a necessary condition towards weak scalability of the entire optimization algorithm, which is why in this article we apply the multigrid method as a preconditioner for the solution of the PDE constraints. This requires to provide a sequence of hierarchically refined discretizations. However, the shape optimization problem is a fine grid problem, which means that the contour of the obstacle has to be representable within the entire grid hierarchy. This leads to

undesired effects and conceptual challenges that have been addressed for instance in Nägel et al. (2015), Siebenborn and Welker (2017), Pinzon et al. (2020).

The finest grid in the hierarchy typically represents a high-resolution discretization of a polygonal shape, which—besides the aforementioned sources for mesh degeneracy—also introduces challenges to the shape optimization. Given that the grid hierarchy stems from an initially coarse grid, the polygonal shape represented by the computational mesh includes geometrical singularities such as edges and corners. As the base level is refined, these singularities become more pronounced. This is an undesired effect that has a negative impact on the discretization, for instance on the mesh quality, which in turn can impact the convergence of the used numerical solvers. Particularly for fluid dynamics applications this is problematic. If the considered domain transformation is too smooth, i.e. the descent directions are chosen in an inappropriate space, it is not possible to remove or form new singularities. The latter is particularly important when using the geometric multigrid method, due to the hierarchical grid structure.

In this work, we propose an approach that is able to identify these regions and adapt the set of admissible transformations F_{adm} as part of the optimization problem. In Onyshkevych and Siebenborn (2021) it is illustrated how adding a nonlinear convection term to the extension model that defines F_{adm} affects forming singularities in optimal shapes. We thus study in this article how the non-linearity can be adjusted according to the shape of the reference domain.

The rest of this article is structured as follows: In Sect. 2 the optimization problem is formulated and the underlying fluid experiment is outlined. Section 3 is devoted to the optimization algorithm and the computation of the reduced gradient via the adjoint method. Subsequently, in Sect. 4 the performance of the proposed method is discussed by presenting numerical tests. In Sect. 5 the numerical scalability of the method is discussed. The article closes with a conclusion in Sect. 6.

2 Problem formulation and mathematical background

The model problem under consideration is sketched in Fig. 1 in a bounded holdall domain $G := \Omega \cup \Omega_{\text{obs}}$, where Ω is assumed to have a Lipschitz boundary. In Ω we consider a stationary, incompressible flow. It surrounds an obstacle Ω_{obs} with variable boundary Γ_{obs} , but fixed volume and barycentric coordinates. Throughout this article, the original setting of the domain will be referred to as the reference configuration or domain.

In this section we present the theoretical background that culminates with the algorithm presented in Sect. 3. The problem is first formulated in terms of classical shape optimization, to be then reformulated as an optimal control problem. Later on, it is pulled back to the reference domain through the method of mappings. The weak form of the extension operator, presented in this section, is used to formulate the augmented Lagrangian. Finally, the Lagrangian is used to obtain the sensitivities necessary for the descent direction and for the approximation to the Hessian used. For an in-depth discussion of the underlying theory we refer the reader, as previously mentioned, to Onyshkevych and Siebenborn (2021),

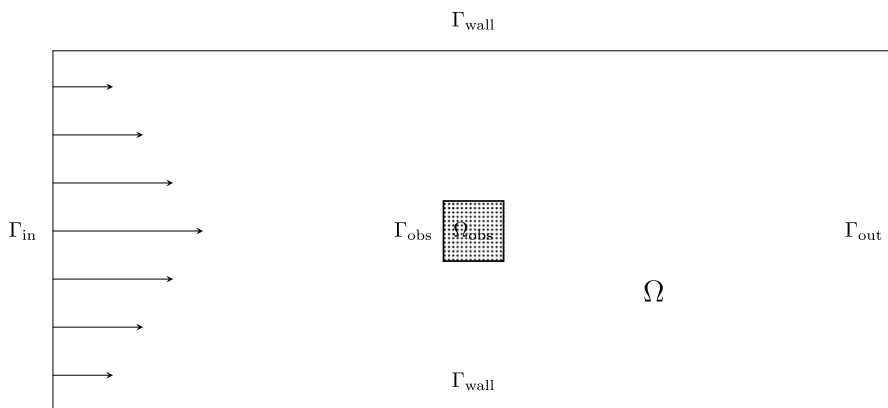


Fig. 1 2d Holdall reference domain of the flow field with square obstacle

Brandenburg et al. (2009), Haubner et al. (2021), the use of the adjoint method in fluid dynamics can be reviewed in Giles and Pierce (2000), Jameson (2003), and Hinze et al. (2009) can be consulted for a mathematical review of the theory used here.

Let $X = L^2(\Gamma_{\text{obs}}) \times L^2(\Omega)$, $0 < \eta_{\text{lb}} < \eta_{\text{ub}}$, $b > 0, \alpha > 0, \theta > 0$ and consider the optimization problem

$$\min_{(u, \eta) \in X} j[y, F(\Omega)] + \frac{\alpha}{2} \|u\|_{L^2(\Gamma_{\text{obs}})}^2 + \frac{\theta}{2} \|\eta - \frac{1}{2}(\eta_{\text{ub}} + \eta_{\text{lb}})\|_{L^2(\Omega)}^2 \quad (3)$$

$$\text{s.t. } e[y, F(\Omega)] = 0 \quad (4)$$

$$F = \text{id} + \mathbf{w} \quad (5)$$

$$\mathbf{w} = S(\eta, u, \Omega) \quad (6)$$

$$\det(DF) \geq b \quad \text{in } \Omega \quad (7)$$

$$\eta_{\text{lb}} \leq \eta \leq \eta_{\text{ub}} \quad \text{in } \Omega \quad (8)$$

$$g(\mathbf{w}) = 0. \quad (9)$$

where $g(\mathbf{w})$ represents geometric constraints. Throughout this work D denotes the Jacobian matrix, while ∇ is the Euclidean gradient operator, and grad is used for the gradient related to the inner product of the corresponding space. S denotes an extension operator, which links the boundary control variable $u \in L^2(\Gamma_{\text{obs}})$ to a displacement field $\mathbf{w} : \Omega \rightarrow \mathbb{R}^d$. Examples of possible choices of S are given and investigated in Haubner et al. (2021). Therein a discussion of the properties of S which guarantee a certain regularity of \mathbf{w} can be found, as well as the resulting regularity of the domain transformation. Here we enrich this operator with an additional control

variable $\eta \in L^2(\Omega)$, which plays the role of a nonlinearity switch. In the following we assume that S is defined such that $\mathbf{w}|_{\Gamma_{\text{in}} \cup \Gamma_{\text{wall}} \cup \Gamma_{\text{out}}} = 0$ almost everywhere.

In the experiment presented here, the geometric constraints require the barycenter and volume to be the origin and constant, respectively. This excludes trivial solutions where the obstacle shrinks to a point or moves towards a position where the objective functional is minimized. Thus, the principal geometric constraints are given by

$$\hat{g}(\mathbf{w}) = \left(\int_{F(\Omega)} 1 \, dx - \int_{\Omega} 1 \, dx, \frac{\int_{F(\Omega)} x \, dx}{\int_{F(\Omega)} 1 \, dx} - \frac{\int_{\Omega} x \, dx}{\int_{\Omega} 1 \, dx} \right)^{\top} \quad (10)$$

This simplifies to

$$g(\mathbf{w}) = \left(\int_{\Omega} \det(DF) - 1 \, dx, \int_{\Omega} F(x) \det(DF) \, dx \right)^{\top}, \quad (11)$$

assuming, without loss of generality, that the barycenter of the reference domain is $0 \in \mathbb{R}^d$ and the volume is precisely retained.

The condition (7) is approximated via a non-smooth penalty term. This results in an approximation via the objective function

$$\begin{aligned} J(y, u, \mathbf{w}, \eta) = & j[y, F(\Omega)] + \frac{\alpha}{2} \|u\|_{L^2(\Gamma_{\text{obs}})}^2 + \frac{\theta}{2} \|\eta - \frac{1}{2}(\eta_{\text{ub}} + \eta_{\text{lb}})\|_{L^2(\Omega)}^2 \\ & + \frac{\beta}{2} \|[b - \det(DF)]^+\|_{L^2(\Omega)}^2. \end{aligned} \quad (12)$$

In contrast to the PDE constraints (5), (6), (7), the geometric constraints (9) are fixed dimensional (here it is $d + 1$ where $d \in \{2, 3\}$). Thus, the multipliers associated to these conditions are not variables in the finite element space but a $d + 1$ -dimensional vector. This is incorporated into the optimization algorithm in the form of an augmented Lagrange approach. This leads to the augmented objective function

$$J_{\text{aug}}(y, u, \mathbf{w}, \eta) := J(y, u, \mathbf{w}, \eta) + \tau \|g(\mathbf{w})\|_2^2, \quad (13)$$

where $\tau > 0$ is a penalty factor for the geometric constraints and $\|\cdot\|_2$ refers to the Euclidean 2-norm since $g(\mathbf{w})$ is finite dimensional. The basic concept of the augmented Lagrange method is to optimize the objective (13). By contrast to a pure penalty method, the geometric constraints (9) are not entirely moved to the objective function, but the corresponding multipliers λ_g are assumed to be approximately known and iteratively updated.

We consider the PDE constraint $e[y, F(\Omega)]$ to be the stationary, incompressible Navier–Stokes equations in terms of velocity and pressure (\mathbf{v}, p) . In the following it is distinguished between PDE solutions defined on the reference domain Ω denoted by (\mathbf{v}, p) and on the transformed domain $F(\Omega)$ as $(\hat{\mathbf{v}}, \hat{p})$. We thus consider

$$-\nu \Delta \hat{\mathbf{v}} + (\hat{\mathbf{v}} \cdot \nabla) \hat{\mathbf{v}} + \nabla \hat{p} = 0 \quad \text{in } F(\Omega) \quad (14)$$

$$\operatorname{div} \hat{\mathbf{v}} = 0 \quad \text{in } F(\Omega) \quad (15)$$

$$\hat{\mathbf{v}} = \mathbf{v}_\infty \quad \text{on } \Gamma_{\text{in}} \quad (16)$$

$$\hat{\mathbf{v}} = 0 \quad \text{on } \Gamma_{\text{obs}} \cup \Gamma_{\text{wall}} \quad (17)$$

$$\hat{p}n - \nu \frac{\partial \hat{\mathbf{v}}}{\partial n} = 0 \quad \text{on } \Gamma_{\text{out}}, \quad (18)$$

where for compatibility it is assumed that $\int_{\Gamma_{\text{in}}} \mathbf{v}_\infty \cdot \mathbf{n} \, ds = 0$ holds for the inflow velocity profile \mathbf{v}_∞ . Notice that the boundaries $\Gamma_{\text{in}}, \Gamma_{\text{out}}, \Gamma_{\text{wall}}$ in (15), (16), (17) are unchanged since the displacement \mathbf{w} is assumed to vanish here. This assumption reflects that the outer boundaries of the experiment domain are not a variable in the optimization problem.

The variational formulation of the PDE (14) to (18) pulled back to the reference domain Ω is given by: Find $(\mathbf{v}, p) \in V \times Q$ such that for all $(\delta_{\mathbf{v}}, \delta_p) \in V_0 \times Q$ it holds

$$\begin{aligned} \int_{\Omega} \nu (D\mathbf{v}(DF)^{-1}) : (D\delta_{\mathbf{v}}(DF)^{-1}) + (D\mathbf{v}(DF)^{-1}\mathbf{v}) \cdot \delta_{\mathbf{v}} \\ - p \operatorname{Tr} (D\delta_{\mathbf{v}}(DF)^{-1}) \det(DF) \, dx = 0 \end{aligned} \quad (19)$$

$$- \int_{\Omega} \delta_p \operatorname{Tr} (D\mathbf{v}(DF)^{-1}) \det(DF) \, dx = 0, \quad (20)$$

where trial and test functions are chosen in

$$\begin{aligned} V &:= \{ \mathbf{v} \in H^1(\Omega, \mathbb{R}^d) : \mathbf{v}|_{\Gamma_{\text{in}}} = \mathbf{v}_\infty, \mathbf{v}|_{\Gamma_{\text{wall}} \cup \Gamma_{\text{obs}}} = 0 \text{ a.e.} \}, \\ V_0 &:= V \text{ with } \mathbf{v}_\infty = 0, \\ Q &:= \left\{ p \in L^2(\Omega) : \int_{\Omega} p \, dx = 0 \right\}. \end{aligned} \quad (21)$$

In the equations above Tr denotes the trace operator, $:$ the double contraction, and \det the determinant.

In the experiment considered in this work the physical part of the objective function (12) is given by the energy dissipation in terms of the velocity \mathbf{v} , thus $y = (\mathbf{v}, p)$ and

$$\hat{j}(\hat{\mathbf{v}}, F[\Omega]) = \nu \int_{F(\Omega)} D\hat{\mathbf{v}} : D\hat{\mathbf{v}} \, dx, \quad (22)$$

which can be pulled back to the reference domain Ω as

$$j(\mathbf{v}, \mathbf{w}) = \nu \int_{\Omega} (D\mathbf{v}(DF)^{-1}) : (D\mathbf{v}(DF)^{-1}) \det(DF) dx. \quad (23)$$

The extension $S(\eta, u, \Omega)$ is defined to be the solution operator of the PDE

$$\operatorname{div}(D\mathbf{w} + D\mathbf{w}^{\top}) + \eta(\mathbf{w} \cdot \nabla)\mathbf{w} = 0 \quad \text{in } \Omega \quad (24)$$

$$(D\mathbf{w} + D\mathbf{w}^{\top})\mathbf{n} = u\mathbf{n} \quad \text{on } \Omega_{\text{obs}} \quad (25)$$

$$\mathbf{w} = 0 \quad \text{on } \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{wall}}. \quad (26)$$

Consider the space

$$W := \{\mathbf{w} \in H^1(\Omega, \mathbb{R}^d) : \mathbf{w}|_{\partial\Omega \setminus \Gamma_{\text{obs}}} = 0 \text{ a.e.}\}. \quad (27)$$

Then the variational formulation of (24), (25), (26) is obtained by: Find $\mathbf{w} \in W$ such that for all $\delta_{\mathbf{w}} \in W$ it holds

$$\int_{\Omega} (D\mathbf{w} + D\mathbf{w}^{\top}) : D\delta_{\mathbf{w}} + \eta(D\mathbf{w} \mathbf{w}) \cdot \delta_{\mathbf{w}} dx = \int_{\Gamma_{\text{obs}}} u\mathbf{n} \cdot \delta_{\mathbf{w}} ds. \quad (28)$$

Finally, we can formulate the approximate optimization problem, which is then solved via the augmented Lagrange approach in Sect. 3, as

$$\min_{(u, \eta) \in X} J_{\text{aug}}(y, u, \mathbf{w}, \eta) \quad (29)$$

$$\text{s.t.} \quad (14) \text{ to } (18) \text{ and } (29) \quad (30)$$

$$F = \operatorname{id} + \mathbf{w} \quad (31)$$

$$\eta_{\text{lb}} \leq \eta \leq \eta_{\text{ub}} \quad \text{in } \Omega \quad (32)$$

$$g(\mathbf{w}) = 0, \quad (33)$$

where the multipliers for conditions (33) are assumed to be known in each iteration.

In order to formulate a gradient-based descent algorithm, we have to compute the sensitivities of the final objective function J_{aug} in (13) with respect to the variables (u, η) . This means to differentiate the chain of mappings

$$(u, \eta) \mapsto \mathbf{w} \mapsto (\mathbf{v}, p) \mapsto J_{\text{aug}}(\mathbf{v}, u, \mathbf{w}, \eta) \quad (34)$$

and obtain the sensitivities in reverse order

$$J_{\text{aug}}(\mathbf{v}, u, \mathbf{w}, \eta) \mapsto (\lambda_{\mathbf{v}}, \lambda_p) \mapsto \lambda_{\mathbf{w}} \mapsto (\lambda_u, \lambda_{\eta}). \quad (35)$$

The derivatives of the mappings mentioned in (34) have been omitted here for brevity, but can be found in Onyshkevych and Siebenborn (2021), Haubner et al. (2021). The differentiability of the mapping is discussed in Sect. 3. Access to the adjoint gradient formulation can be obtained via the corresponding Lagrangian, which is given by

$$\begin{aligned}
 \mathcal{L}(\mathbf{w}, \mathbf{v}, p, u, \eta, \lambda_{\mathbf{w}}, \lambda_{\mathbf{v}}, \lambda_p, \lambda_{\text{bc}} \lambda_{\text{vol}}) = & \frac{\nu}{2} \int_{\Omega} (D\mathbf{v}(DF)^{-1}) : (D\mathbf{v}(DF)^{-1}) \det(DF) \, dx \\
 & + \frac{\alpha}{2} \int_{\Gamma_{\text{obs}}} u^2 \, ds + \frac{\beta}{2} \int_{\Omega} ([\eta_{\text{det}} - \det(DF)]^+)^2 \, dx + \frac{\theta}{2} \int_{\Omega} \left[\eta - \frac{1}{2}(\eta_{\text{ub}} + \eta_{\text{lb}}) \right]^2 \, dx + \tau \|g(\mathbf{w})\|_2^2 \\
 & + \int_{\Omega} (\nu [D\mathbf{v}(DF)^{-1}] : [D\lambda_{\mathbf{v}}(DF)^{-1}] + [D\mathbf{v}(DF)^{-1}] \cdot \lambda_{\mathbf{v}} - p \operatorname{Tr} [D\lambda_{\mathbf{v}}(DF)^{-1}] \det(DF) \, dx \\
 & - \int_{\Omega} \lambda_p \operatorname{Tr} [D\mathbf{v}(DF)^{-1}] \det(DF) \, dx + \int_{\Omega} [(D\mathbf{w} + D\mathbf{w}^{\top}) : D\lambda_{\mathbf{w}} + \eta(D\mathbf{w} \cdot \mathbf{w})] \, dx \\
 & - \int_{\Gamma_{\text{obs}}} \mathbf{u}\mathbf{n} \cdot \lambda_{\mathbf{w}} \, ds + \lambda_{\text{bc}} \cdot \int_{\Omega} F(x) \det(DF) \, dx + \lambda_{\text{vol}} \int_{\Omega} [\det(DF) - 1] \, dx
 \end{aligned} \tag{36}$$

under the assumption that the barycenter of Ω is $0 \in \mathbb{R}^d$.

From the Lagrangian (36) the adjoint Navier–Stokes equations follow as: Find $(\lambda_{\mathbf{v}}, \lambda_p) \in V_0 \times Q$ such that for all $(\delta_{\mathbf{v}}, \delta_p) \in V_0 \times Q$ it holds

$$\begin{aligned}
 & \int_{\Omega} \nu (D\lambda_{\mathbf{v}}(DF)^{-1}) : (D\delta_{\mathbf{v}}(DF)^{-1}) + (D\delta_{\mathbf{v}}(DF)^{-1} \mathbf{v}) \cdot \lambda_{\mathbf{v}} \\
 & + (D\mathbf{v}(DF)^{-1} \delta_{\mathbf{v}}) \cdot \lambda_{\mathbf{v}} - \lambda_p \operatorname{Tr} (D\delta_{\mathbf{v}}(DF)^{-1}) \det(DF) \, dx = 0
 \end{aligned} \tag{37}$$

$$- \int_{\Omega} \delta_p \operatorname{Tr} (D\lambda_{\mathbf{v}}(DF)^{-1}) \det(DF) \, dx = 0, \tag{38}$$

The adjoint displacement equation is obtained by: Find $\lambda_{\mathbf{w}} \in W$ such that for all $\delta_{\lambda_{\mathbf{w}}} \in W$ it holds

$$\int_{\Omega} (D\lambda_{\mathbf{w}} + D\lambda_{\mathbf{w}}^{\top}) : D\delta_{\lambda_{\mathbf{w}}} + \eta(D\mathbf{w} \lambda_{\mathbf{w}}) \cdot \delta_{\lambda_{\mathbf{w}}} \, dx = R(\mathbf{w}, \mathbf{v}, p, \lambda_{\mathbf{v}}, \lambda_p). \tag{39}$$

In (39) R denotes the derivative of the Lagrangian (36) w.r.t. \mathbf{w} . This is obtained after straightforward computations and omitted here for the sake of brevity. Finally, the reduced gradient is obtained as: Find $(\gamma, \kappa) \in X$ such that for all $(\delta_u, \delta_{\eta}) \in X$ it holds

$$\int_{\Gamma_{\text{obs}}} \gamma \delta_\gamma + \alpha u \delta_u - \lambda_w \cdot n \delta_u \, ds = 0, \quad (40)$$

$$\int_{\Omega} \kappa \delta_\eta + \theta \left(\eta - \frac{1}{2}(\eta_{\text{ub}} + \eta_{\text{lb}}) \right) \delta_\eta - (D\mathbf{w} \cdot \mathbf{w}) \cdot \lambda_w \delta_\eta \, ds = 0. \quad (41)$$

With the sensitivity equations (40), (41) we are now prepared to apply a descent method.

3 Optimization algorithm

In Sect. 2 we present the approximate optimization problem (29), (30), (31), (32), (33), which is solved via the augmented Lagrange approach shown in Algorithm 1. An initial guess is given to the Lagrange multipliers λ_g associated to the geometrical constraints. These in turn are iteratively updated in each optimization step subject to the condition that the norm of the defect of the geometrical constraints is smaller than a prescribed tolerance $\epsilon_g > 0$. In Algorithm 1 only the problem-dependent parameters have to be provided, which are not the ones immediately related to the extension operator $S(\eta, u, \Omega)$. The parameters that define the set of admissible shapes, e.g. η , are simultaneously optimized. This is a significant improvement to previous approaches, where the parameters had to be manually determined, e.g. Blauth (2021), Schulz and Siebenborn (2016).

Most of the computational time is consumed for solving the PDE systems presented in Sect. 2. This is carried out by Algorithm 3 in a block-wise manner, where the output consists of the new displacement field to update the transformation (5), as well as the results of the reduced gradient $\text{grad } u^{k,\ell}$, $\text{grad } \eta^{k,\ell}$, which will be further used to obtain updates for the current control and extension factor, $u^{k,\ell+1}$, $\eta^{k,\ell+1}$, respectively. A reference for the gradient method can be found for instance in (Hinze et al. 2009, p. 94).

Algorithm 1 Augmented Lagrange Outer Optimization Algorithm**Require:** $\lambda_g, \lambda_{\text{inc}} > 0, 0 \leq \eta_{\text{lb}} \leq \eta \leq \eta_{\text{ub}}, b > 0, \epsilon_g > 0, \tau > 0, \tau_{\text{inc}} > 0, \epsilon_{\text{outer}} > 0, \epsilon_{\text{inner}} > 0$

```

1: repeat
2:    $\ell \leftarrow 0$ 
3:   repeat
4:      $(w, \text{grad } u^{k,\ell}, \text{grad } \eta^{k,\ell}) \leftarrow \text{REDUCED\_GRADIENTS}(g_{\text{def}}, u^{k,\ell}, \eta^{k,\ell})$ 
5:      $(u^{k,\ell+1}, \eta^{k,\ell+1}) \leftarrow \text{LBFGS-B}(m, \ell, g_{\text{def}}, \tau, u^{k,\ell}, \eta^{k,\ell}, \text{grad } u^{k,\ell}, \text{grad } \eta^{k,\ell})$ 
6:      $g_{\text{def}} \leftarrow g(w) - g(0)$ 
7:      $\ell \leftarrow \ell + 1$ 
8:   until  $\left\| \left( \text{grad } u^{k,\ell-1}, P_{(\eta_{\text{lb}}, \eta_{\text{ub}})} \left( \eta^{k,\ell-1} - \text{grad } \eta^{k,\ell-1} \right) - \eta^{k,\ell-1} \right) \right\|_X < \epsilon_{\text{inner}}$ 
9:   if  $\|g_{\text{def}}\|_2 < \epsilon_g$  then
10:      $\tau \leftarrow \tau_{\text{inc}} \tau$ 
11:   else
12:      $\lambda_g \leftarrow \lambda_g + \lambda_{\text{inc}} g_{\text{def}}$ 
13:   end if
14:    $k \leftarrow k + 1$ 
15: until  $k \geq 1$  and  $\|u^{k,\ell} - u^{k-1,\ell}\|_{L^2(\Gamma_{\text{obs}})} < \epsilon_{\text{outer}}$ 

```

The objective function is non-differentiable due to the presence of the positive part mapping in R in (39), which is discussed in depth in Haubner et al. (2021). Moreover, a discussion on quasi-Newton methods for semi-smooth objective functions and can be found in Mannel and Rund (2020).

Algorithm 2 Limited Memory BFGS Algorithm with Box Constraints Inner Iteration

```

1: function LBFGS-B( $m, \ell, g_{\text{def}}, \tau, w, u^\ell, \eta^\ell, \text{grad } u^\ell, \text{grad } \eta^\ell$ )
2:   Determine  $\chi_\eta$  according to (43) and  $(\cdot, \cdot)_{\hat{X}}$  according to (45)
3:    $q = (q_1, q_2) \leftarrow (\text{grad } u^\ell, \text{grad } \eta^\ell)$ 
4:   if  $\ell > 0$  then
5:      $s^{\ell-1} \leftarrow (u^\ell, \eta^\ell) - (u^{\ell-1}, \eta^{\ell-1})$ 
6:      $z^{\ell-1} \leftarrow (\text{grad } u^\ell, \text{grad } \eta^\ell) - (\text{grad } u^{\ell-1}, \text{grad } \eta^{\ell-1})$ 
7:     for  $i = \ell - 1, \dots, \max\{\ell - m, 0\}$  do
8:        $\alpha_i \leftarrow \rho_i(s^i, q)_{\hat{X}}$ 
9:        $q \leftarrow q - \alpha_i z^i$ 
10:    end for
11:     $q \leftarrow \frac{(s^{\ell-1}, z^{\ell-1})_{\hat{X}}}{(z^{\ell-1}, z^{\ell-1})_{\hat{X}}} q$ 
12:    for  $i = \max\{\ell - m, 0\}, \dots, \ell - 1$  do
13:       $q \leftarrow q + (\alpha_i - \rho_i(y^i, q)_{\hat{X}}) s^i$ 
14:    end for
15:  end if
16:  return  $(u + q_1, P_{(\eta_{\text{lb}}, \eta_{\text{ub}})}(\eta + q_2))$ 
17: end function

```

The use of box-constraints for the extension factor η make it necessary to implement the BFGS method similarly to what can be found in Byrd et al. (1995), from which Algorithm 2 is partly inspired. For the box-constrained limited memory BFGS method, we introduce the indicator function for the condition $\eta_{\text{lb}} \leq \eta \leq \eta_{\text{ub}}$ as

$$\chi_\eta(x) := \begin{cases} 1, & \text{if } \eta_{\text{lb}} \leq \eta(x) - \sigma \text{grad } \eta(x) \leq \eta_{\text{ub}} \\ 0, & \text{else} \end{cases} \quad (42)$$

for some small $\sigma > 0$. Recall that the canonical inner product on X is given as

$$_X = (u_1, u_2)_{L^2(\Gamma_{\text{obs}})} + (\eta_1, \eta_2)_{L^2(\Omega)}. \quad (43)$$

This is now modified to take the active box-constraints into account by introducing χ_η into the second term and thereby reducing the integration to the region of inactive constraints

$$\hat{X} = (u_1, u_2)_{L^2(\Gamma_{\text{obs}})} + (\eta_1, \chi_\eta \eta_2)_{L^2(\Omega)} \quad (44)$$

Eq. (44) defines the inner product appearing in lines 8, 11, 13 of Algorithm 2. Whereas, in line 16 the operator P refers to the projection with respect to the box constraints on η .

Conceptually the optimization scheme presented consists of outer and an inner iterations. The outer iteration, seen in Algorithm 1, updates either λ_g or the penalty factor τ by increment factors λ_{inc} and τ_{inc} , respectively. In each cycle of the inner loop, a complete optimization is solved using BFGS updates as seen in Algorithm 2.

Algorithm 3 Computation of Reduced Gradient

```

1: function GRADIENT( $g_{\text{def}}, u, \eta$ )
2:    $u \mapsto w$  via (29)
3:    $w \mapsto (v, p)$  via (20) and (21)
4:    $(v, p, w) \mapsto (\lambda_v, \lambda_p)$  via (40)
5:    $(v, p, w, \lambda_v, \lambda_p, \lambda_g) \mapsto \lambda_w$  via (40)
6:    $\lambda_w \mapsto (\gamma, \kappa)$  via (41) and (42)
7:   return ( $w, \gamma, \kappa$ )
8: end function
  
```

4 Shape optimization applications

In this section we present shape optimization applications with the incompressible, stationary Navier–Stokes equations as state equation. The purpose of the featured case studies in this section is to show the application of the algorithm presented in Sect. 3, which includes the effect of the nonlinearity control variable η on the extension operator S . The obstacle shape deformations demonstrate the algorithm’s capabilities at the detection, smoothing and creation of domain singularities such as tips and edges. Aspects of the multigrid preconditioner’s effects are discussed. Moreover, a grid independence study illustrates that the optimal shape is reached regardless of the number of refinements in the grid hierarchy. The latter result is a fundamental stepping stone towards a scalable parallel implementation of the methodology proposed in Sect. 5.

The flow tunnel is depicted as the holdall domain in Fig. 1 with

$$G_{2d} = (-7, 7) \times (-3, 3) \text{ and } G_{3d} = (-7, 7) \times (-3, 3) \times (-3, 3)$$

for the 2d and 3d cases respectively, taking into account that for the 3d case the obstacle has a spherical shape. Thus, in 2d we have $\Omega_{\text{obs}} = (-0.5, 0.5)^2$ and $\Omega_{\text{obs}} = \{x \in \mathbb{R}^3 : \|x\|_2 < 1\}$ in 3d, respectively.

The boundary conditions at the inflow boundary Γ_{in} are set as

$$v_{\infty} = \left(\max \left\{ 0, \prod_{i=2}^d \cos\left(\frac{\pi|x_i|}{\delta}\right) \right\}, 0, \dots, 0 \right) \in \mathbb{R}^d$$

with δ the diameter of the flow tunnel. The side length of the square obstacle is $d = 1$, whereas the radius of the sphere in the 3d case is $r = 0.5$. The simulations are performed using UG4 Vogel et al. (2013). We expand UG4 through its C++ based

plugin functionality. The code used for the studies here presented can be consulted at the online repository in Pinzon and Siebenborn (2021). The 2d and 3d grids are generated using the GMSH toolbox Geuzaine and Remacle (2009).

4.1 2d Results

In this section we present 2d simulations for a flow with viscosity $\nu = 0.03$. All PDEs are discretized using a P_1 approximation, except for the Navier–Stokes equations and its adjoint which are solved with a stable $P_2 - P_1$ finite element discretization. For this example, η has initial value of 0.5 and box constraints are $0 \leq \eta \leq 1.0$ and $b = 0.001$. The grid consists of 421,888 triangular elements, with 5 refinement levels. Figure 2 shows results for the optimization of a square obstacle subject to an incompressible, stationary flow. The reference configuration with the extension factor η and optimal displacement field are shown together with the transformed domain and a closeup of the front tip where the element edges are depicted. Regarding the reference configuration, it can be seen that the extension factor approaches the imposed values of the box-constraints at two places, the corners of the square and the sections where new singularities have to be created. If we recall the weak form of the extension factor (28), η controls the nonlinearity in each element. Given that the same initial value of η is set for all elements at the beginning of the simulation and the θ -term in (12) penalizes the deviation from the average of $\frac{1}{2}(\eta_{ub} + \eta_{lb})$, the different extension factor values, particularly the ones close to the obstacle's surface Γ_{obs} , show that equation (28) adapts depending on the current iterate for the displacement field \mathbf{w} . This ensures that the \mathbf{w} promotes both the generation of new

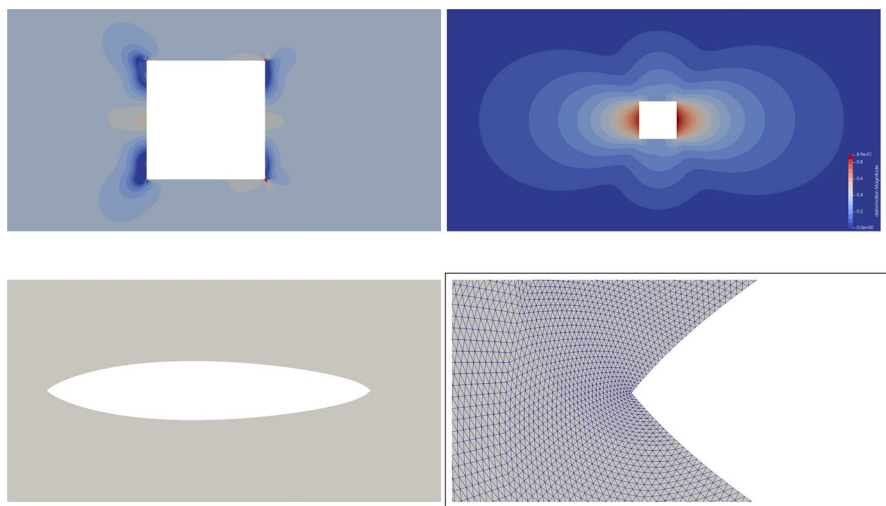


Fig. 2 At the top, the reference configuration is shown with the optimal η (left) and $\mathbf{w} = S(\mathbf{u})$. At the bottom, the transformed grid $F(\Omega)$ with resulting singularities (left) is shown, altogether with a zoom on the singularity where mesh quality is preserved due to the choice of S

non-smooth points on the boundary, as well as the smoothing of such points introduced by the choice of the reference domain, i.e. the four corners of the box inclusion Ω_{obs} . This can be observed in Fig. 2, where high valued displacements are present at the sections where the tips and corners are generated or smoothed. Which in turn leads to achieving large deformations $F(\Omega)$ without loss of convergence of the iterative solvers.

In Sect. 2 we already mention that explicit mesh deformations are avoided. This comes from the fact that all optimization steps are solved on the reference domain through the method of mappings. Therefore we speak of obtaining an optimal deformation field F , which is used to transform the domain $\Omega \mapsto F(\Omega)$ according to (5). This is used to obtain the optimal shape, as in Fig. 2. The transformed domain shows the smoothed corners and the generated front and back surface singularities, which are in accordance to the previously mentioned properties of η and \mathbf{w} . However, throughout the optimization process the proposed algorithm does not require the nodal positions to be redefined, since the reference grid transformation is only performed as part of the post-processing and not of the optimization. The close-up corresponds to the front singularity with respect to the direction of flow. Figure 2 also shows that elements around the generated tip show no distortion and no significant loss of quality. This stems from both the effect of the nonlinear term in the extension operator S and the imposed upper bound b on the determinant of the deformation gradient $\det(DF)$, given in (7). The latter condition is what preserves local injectivity, thus avoiding the loss of mesh quality. In Fig. 2 this is shown as the absence of collapsed or overlapping elements, as is previously mentioned, even for the elements that clearly undergo large deformations, i.e. the ones that conform the generated tips and the smoothed square corners. Moreover, in Sects. 4.3 and 5 this can be understood as a mesh independent preservation of the geometrical and numerical convergence in terms of the final optimal shape achieved and the total iteration counts of the iterative solvers.

On the other hand, the extension equation adapts where the tips have to be created to reach an optimal value of the objective function. This is illustrated through the changing value of the nonlinearity switch η on each optimization step. Figure 3 shows the plot of η over the reference domain compared to the domain transformed by the displacement field \mathbf{w} .

At the start of the simulation the extension equation is already adapted to find the corners of the reference configuration, this is shown as the concentrated value of η . As the obstacle's initial singularities are removed, necessary ones are created. This causes a concentration of the extension factor at the reference configuration locations where new geometrical singularities have to be created. Afterwards, the optimization scheme works towards smoothing the obstacle's surface, therefore η goes through no major concentration values across the grid, as can be seen in step 74 of Fig. 3.

The distribution of η across the grid has to be compared against the transformed grid. Given that the Lagrange multipliers are yet to converge, the initial steps can incur in violations of the geometrical constraints. This can be seen as the highly deformed shapes at the initial steps of Fig. 3. However, as the algorithm performs the multipliers' update, as in Algorithm 1 line 12, the

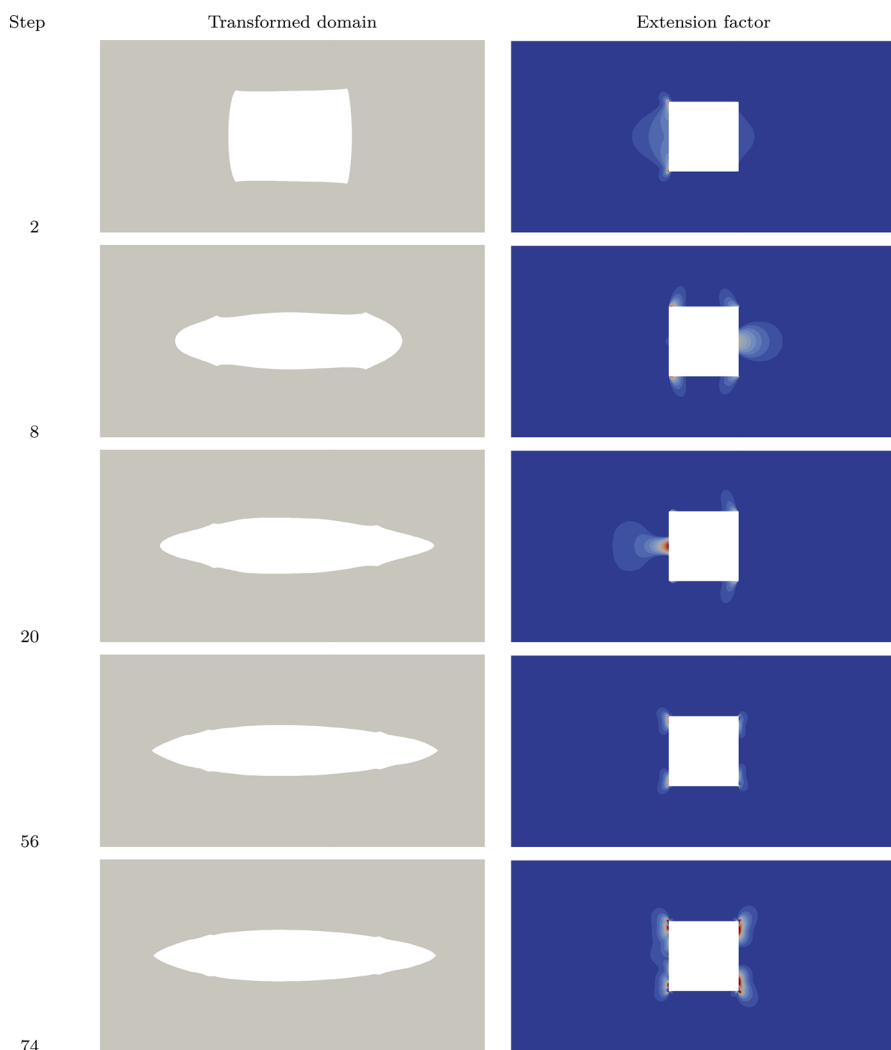


Fig. 3 Transformation of the reference domain by the application of the deformation field compared to the accumulation of the extension factor, given for steps 2, 8, 20, 56, 74

geometrical constraints are fulfilled according to the prescribed ϵ_g and the new obstacle surface's singularities are formed. Moreover, since the reference configuration singularities are identified at the initial optimization steps, the necessary smoothing is carried out until the simulation converges or the maximum number of steps is reached. This can be seen comparing the last 2 steps of Fig. 3.

4.2 3d Results

3d Results for the optimization of a unit-diameter sphere are presented here. For these results, 4 levels of refinements are used with up to 12,296,192 tetrahedral elements, while the obstacle's surface consists of 54,784 triangular elements. The viscosity is set to $\nu = 0.1$, with a discretization scheme as in Sect. 4.1 where, in contrast to the 2d case, $P_1 - P_1$ mixed elements are used for the Navier–Stokes equations and its adjoint. Regarding the extension equation, η has initial value of 30 and the box constraints are $0 \leq \eta \leq 60.0$. A pressure projection stabilization term is used for the mixed finite element approximation, as given in Elman et al. (2014).

The discretized domain representing the 3d obstacle is shown in Fig. 4. Both the base level (bold blue lines) and highest refinement level are presented. As mentioned in Sect. 1, we investigate the application of the geometric multigrid method as a preconditioner in shape optimization. This implies that we strive to maintain the base level as coarse as possible, as can be seen by the underrepresented sphere shown; with the idea of solving the coarsest problem with a direct method as quickly as possible. While this is ideal for the usage and convergence of the geometric multigrid method, it has some undesired effects. As can be seen, the several refinements introduced by the creation of the hierarchical grid levels do not necessarily introduce a smoothing of the obstacle's surface. The refinements are limited to subdividing the triangular faces present on Γ_{obs} , while the edges from the base grid remain.

The results after 61 steps are shown in Fig. 5. Non-smooth points, i.e. the two tips, are generated on the front and back of Γ_{obs} with respect to the direction of flow. This

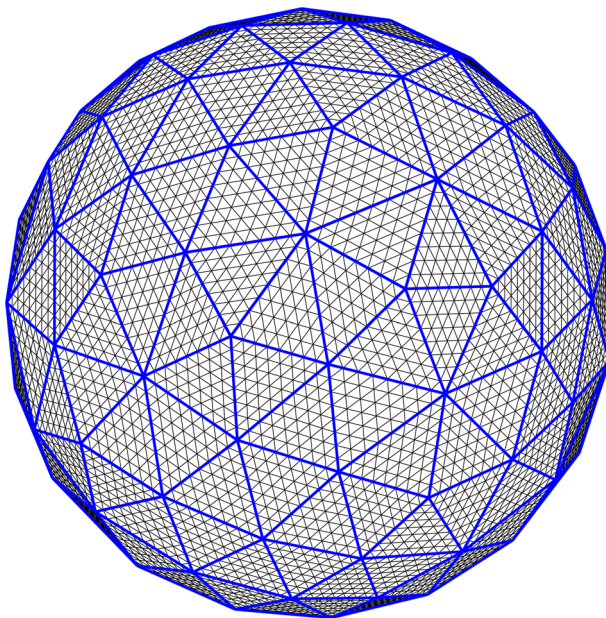


Fig. 4 3d Highest grid level is shown compared to the base level (bold lines) for the unit-diameter sphere obstacle

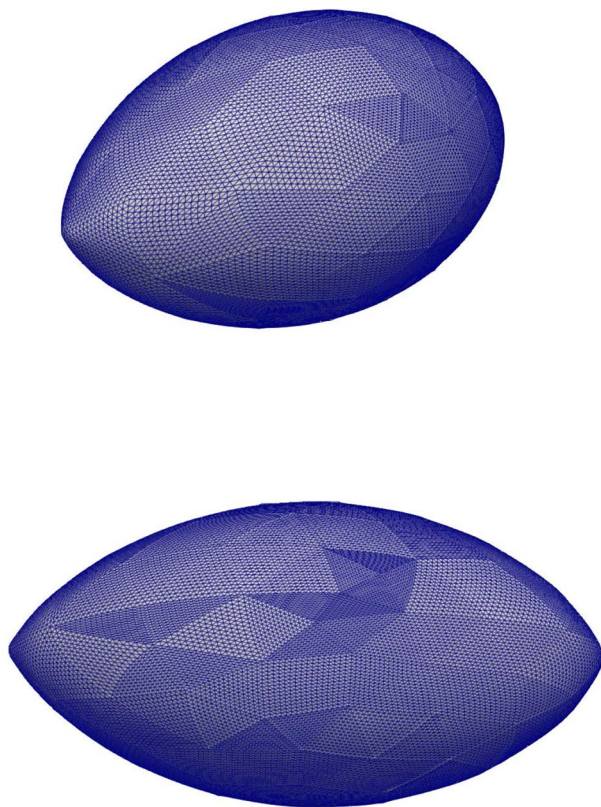


Fig. 5 3d optimal shape on the highest level of refinement in flow with large viscosity

is comparable to the optimal shape obtained for the 2d case in Fig. 2. The effects of the grid hierarchy can be seen as the remaining edges of the super-elements.

4.3 Grid independence study

In order for the proposed optimization scheme to be scalable in terms of time-to-solution to very high numbers of DoFs, it is necessary for the obtained obstacle shape to be independent of the initial level of refinement. In other words, besides the scalability of the finite element building blocks of the optimization algorithm, the overall convergence of the objective function has to be mesh independent. This can be understood as obtaining the same optimized shape after a given number of outer iterations in Algorithm 1, with the necessary surface singularities appearing at approximately the same locations. Therefore, in this section we provide results for a comparative study between different levels of refinement.

The grid used in this section is formed by 412 triangular elements and refined to 1,687,552 elements. The results shown go from 2 to 6 refinement levels. The simulations are set with viscosity $\nu = 0.1$. An equal number of 400 optimization

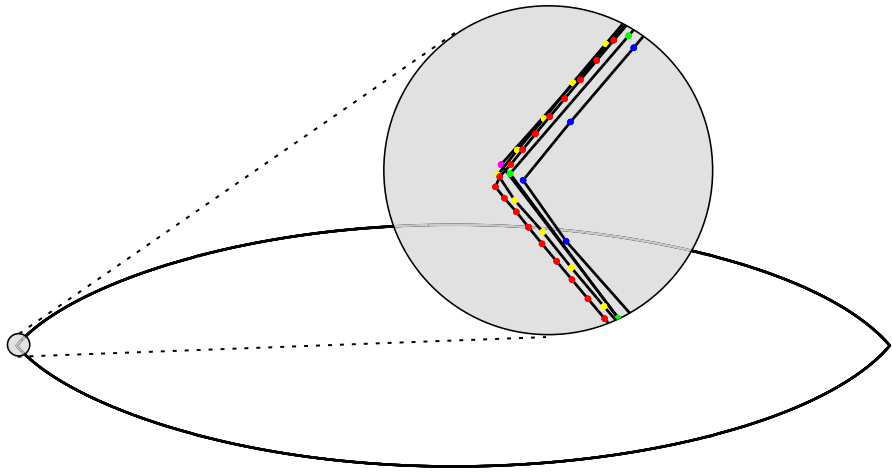


Fig. 6 Optimal displacement field w , after 400 optimization steps, applied to the reference shape for several levels of refinement, indicated by colored nodes

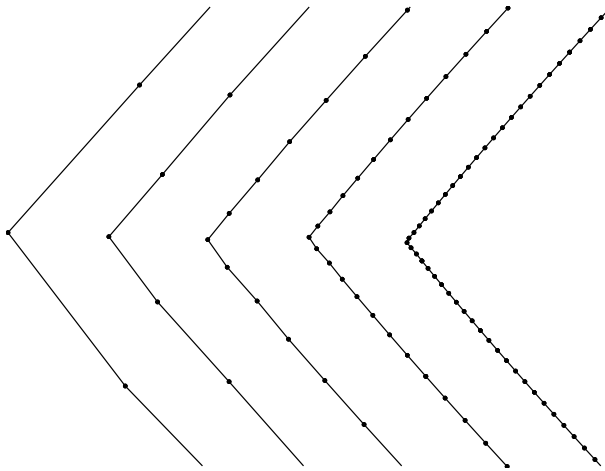


Fig. 7 Zoom into the tips of the deformed shapes of the experiment shown in Fig. 6. Comparison of generated boundary singularities for 2, 3, 4, 5, and 6 levels of refinement (from left to right). Shapes are aligned with fixed interspace for better comparability

steps is run for all grids, to have an adequate comparison point. Figure 6 shows the superimposed contours of the obstacle for 2, 3, 4, 5, and 6 levels of refinement, indicated by the colored nodes. A magnification is used on the front tip, to emphasize that all tips appear on the same location, with slight differences owing to the discretization error introduced by different element sizes. In addition to this results, Fig. 7 shows a side-by-side comparison of the tips of the aforementioned

refinement levels. This indicates, that the singularities on the obstacle surface generated by algorithm Algorithm 1, are grid independent.

Moreover, in Fig. 8 the objective function plots for 2 different refinement levels are shown. With the same viscosity as in Figs. 6 and 7, the simulations are set to run for 1000 steps. The purpose is to demonstrate that the achieved minimal value is independent of the geometry. We thus choose this large number of optimization steps regardless of any tolerance ϵ_{outer} . The value of (13) (in blue) is compared against the Euclidean norm of the Lagrange multipliers (in green) for each refinement, while the update of the multipliers is signaled by the dashed lines (dark blue). It is evident that before the convergence of the multipliers, the optimization process is local, which is why differences between the two plots with respect to the objective function value are present. In Algorithm 1, the condition for the update of the aforementioned Lagrange multipliers is mentioned. This is related to the set tolerance ϵ_g . Given that the two geometries are different due to the level of refinement, the fulfillment of the geometrical tolerance is not necessarily achieved in the same optimization steps. Which in turn, as seen in Algorithm 1, has an effect as to when the multipliers are updated. Nevertheless, as previously mentioned in this section, the objective function converges altogether with the norm of the multipliers, as seen by comparing the plots in both refinement cases presented in Fig. 8. It can also be seen that in most cases, a significant update of the Lagrange multipliers is accompanied by a substantial jump, negative or positive, in the value of the objective function. This is signaled by the intersection data points between

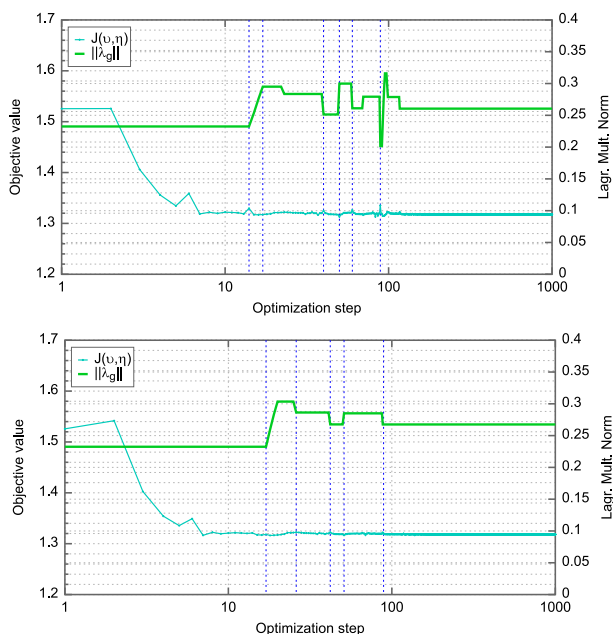


Fig. 8 Objective function plot for 3 (top) and 4 (bottom) refinements. Green line shows norm of geometric Lagrange multipliers. Dashed blue vertical lines indicate λ_g update in augmented Lagrange algorithm

the changes in the norm level of the multipliers $\|\lambda_g\|$, the jumps of the objective, and the marked update steps.

5 Algorithmic scalability

In this section we present weak scalability results for the 2d case presented in Sect. 4.1. These were carried out at *HLRS* using the modern *HPE Hawk* supercomputer. It consists of 5632 dual-socket nodes with the AMD EPYC 7742 processor. Each node has a total of 128 cores and 256GB of memory. The machine presents a 16-node hypercube connection topology, therefore the core counts are aimed at maximizing hypercube use, without significantly reducing the bandwidth. The grid partitioning is based on ParMETIS Karypis et al. (2013).

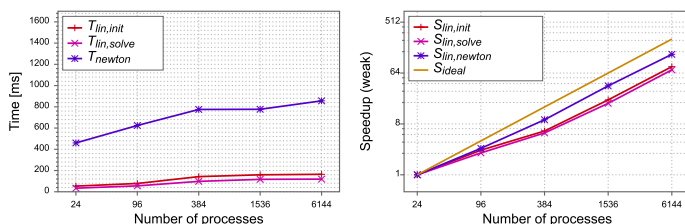
Figure 9 shows accumulated wallclock times, gained speedup relative to 24 cores and iteration counts for the first three optimization steps. These results are shown for the nonlinear extension operator (28), state (14), (15), (16), (17), (18), and the adjoint displacement (39). A P_1 finite element discretization is used for extension operator and its adjoint equation, while mixed $P_1 - P_1$ shape functions are used for the state equation Navier–Stokes equations. The nonlinear problems are solved using Newton’s method altogether with a BiCGStab solver for the underlying linearizations. The linear solver is preconditioned with the geometric multigrid, which uses a V-cycle, 3 pre- and post-smoothing steps, and an LU base solver gathered on a single core. The error reduction is set to an absolute of 10^{-14} and relative 10^{-8} for the nonlinear solver, and 10^{-12} and 10^{-3} for the linear solvers. The purely linear problem has a relative and absolute reduction of 10^{-16} . A Jacobi smoother is used within the geometric multigrid for the extension equation and the derivative, whereas the Navier–Stokes equation solver features an ILU smoother [see for instance Wittum (1989)]. The results presented start at 24 cores, with a fourfold increase for each mesh refinement.

The studies show scalability and speedup for up to 6144 cores and more than 27 million triangular elements. Given that mesh refinements are performed for each core count increase, a different geometrical problem is solved therefore differences in iteration counts are expected. However, even for a significant increase in the number of geometric elements the iteration counts for the linear problems remain within moderate bounds. Moreover, it is important to point out that the total number of DOFs solved within the presented PDEs in Fig. 9 increases from about 783k to 189 million. While the total number of DOFs solved in one optimization step is close to 300 million.

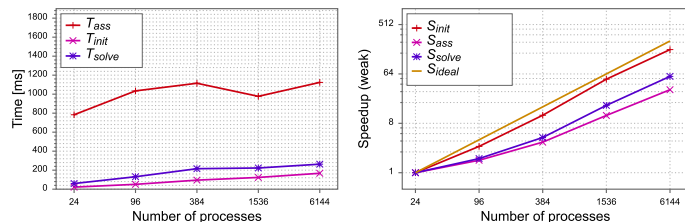
Together with the grid independence study for the outer optimization routine we thus obtain weak scalability of the overall method.

6 Conclusion

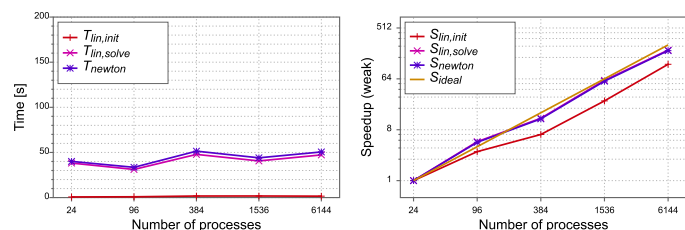
In this article we presented an optimization methodology which relies on the self-adaption of the extension operator within the method of mappings. The results show that large deformations with respect to the reference configuration are possible while



(a) Extension equation nonlinear problem



(b) Adjoint displacement linear problem



(c) State equation nonlinear problem

Procs	Refs	NumElems	Adjoint Displacement (linear solver)	Displacement Field (linear/nonlinear solver)	State Equation (linear/nonlinear solver)
24	4	105472	56	9/12	238/12
96	5	421888	70	9/12	199/12
384	6	1687552	77	9/12	211/13
1536	7	6750208	77	9/12	194/13
6144	8	27000832	70	8/10	233/12

(d) Accumulated iteration counts for the weak scaling study

Fig. 9 Weak scaling: For the first three optimization steps, the accumulated wallclock time is shown for: **a** the nonlinear extension equation, **b** the derivative of the objective function with respect to the displacement field. In **c**, the accumulated iteration counts are presented for the geometric multigrid preconditioned linear solver of the shape derivative, the number of Newton steps and linear iterations necessary to solve the extension equation and its linearization

preserving mesh quality. This has been studied in situations where singularities during the shape optimization process have to be smoothed out and new ones generated. It has been demonstrated that these two effects are particularly important to be tackled for applications of hierarchical multigrid solvers when experiments from fluid dynamics are considered.

The method's scalability and grid independency have been illustrated with the results of Sects. 4.3 and 5. Grid independence is necessary for applications where a high level of refinement is needed, since it guarantees that the same optimal shape is obtained regardless of the number of elements. This becomes particularly important for the weak scalability, where the grid is refined on each core count increase. The results shown in Fig. 9, in combination with the ones of Figs. 6 and 7, establish a proof of concept for industrial applicability, where a high number of DOFs are expected. Overall, in this article we have presented an algorithm towards scalable shape optimization for large scale problems with the potential to work reliably also in complex geometric situations.

Acknowledgements Computing time on the supercomputer *Hawk* at HLRS under the grant ShapeOpt-CompMat (ACID 44171, Shape Optimization for 3d Composite Material Models) is gratefully acknowledged. The current work is part of the research training group “Simulation-Based Design Optimization of Dynamic Systems Under Uncertainties” (SENSUS) funded by the state of Hamburg under the aegis of the Landesforschungsförderungs-Projekt LFF-GK11. The authors acknowledge the support by the Deutsche Forschungsgemeinschaft (DFG) within the Research Training Group GRK 2583 “Modeling, Simulation and Optimization of Fluid Dynamic Applications”.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Allaire G, Dapogny C, Jouve F (2021) Chapter 1—shape and topology optimization. In: Bonito A, Nochetto RH (eds) Geometric partial differential equations—part II, handbook of numerical analysis, vol 22. Elsevier, Amsterdam, pp 1–132. <https://doi.org/10.1016/bs.hna.2020.10.004>
- Baker AH, Falgout RD, Kolev TV, Yang UM (2011) Multigrid smoothers for ultra-parallel computing. *SIAM J Sci Comput* 33:2864–2887
- Blauth S (2021) Nonlinear conjugate gradient methods for PDE constrained shape optimization based on steklov-poincaré-type metrics. *SIAM J Optim* 31(3):1658–1689. <https://doi.org/10.1137/20M1367738>
- Brandenburg C, Lindemann F, Ulbrich M, Ulbrich S (2009) A continuous adjoint approach to shape optimization for Navier Stokes flow. In: Kunisch K, Leugering G, Sprekels J, Tröltzsch F (eds) Optimal control of coupled systems of partial differential equations, vol 160. Birkhäuser, Basel, pp 35–56
- Byrd RH, Lu P, Nocedal J, Zhu C (1995) A limited memory algorithm for bound constrained optimization. *SIAM J Sci Comput* 16(5):1190–1208. <https://doi.org/10.1137/0916069>

- Delfour MC, Zolésio JP (2001) Shapes and geometries: metrics, analysis, differential calculus, and optimization. *Advances in design and control*, vol 22, 2nd edn. SIAM, Philadelphia
- Dokken JS, Funke SW, Johansson A, Schmidt S (2019) Shape optimization using the finite element method on multiple meshes with nitsche coupling. *SIAM J Sci Comput* 41(3):A1923–A1948
- Dokken JS, Mitusch SK, Funke SW (2020) Automatic shape derivatives for transient PDEs in FEniCS and Firedrake
- Elman H, Silvester D, Wathen A (2014) Finite elements and fast iterative solvers with applications in incompressible fluid dynamics, vol 1. Oxford Science Publications
- Etling T, Herzog R, Loayza E, Wachsmuth G (2018) First and second order shape optimization based on restricted mesh deformations. *SIAM J Sci Comput* 42(2):A1200–A1225
- Fischer M, Lindemann F, Ulbrich M, Ulbrich S (2017) Fréchet differentiability of unsteady incompressible Navier–Stokes flow with respect to domain variations of low regularity by using a general analytical framework. *SIAM J Control Optim* 55(5):3226–3257. <https://doi.org/10.1137/16M1089563>
- Gangl P, Laurain A, Meftahi H, Sturm K (2015) Shape optimization of an electric motor subject to non-linear magnetostatics. *SIAM J Sci Comput* 37(6):B1002–B1025
- Garcke H, Hinze M, Kahle C (2016) A stable and linear time discretization for a thermodynamically consistent model for two-phase incompressible flow. *Appl Numer Math* 99:151–171
- Geuzaine C, Remacle JF (2009) Gmsh: a 3-D finite element mesh generator with built-in pre- and post-processing facilities. *Int J Numer Methods Eng* 79(11):1309–1331. <https://doi.org/10.1002/nme.2579>
- Giles MB, Pierce NA (2000) An introduction to the adjoint approach to design. *Flow Turbul Combust* 65(3–4):393–415
- Gmeiner B, Köstler H, Stürmer M, Rude U (2014) Parallel multigrid on hierarchical hybrid grids: a performance study on current high performance computing clusters. *Concurr Comput Pract Exp* 26(1):217–240
- Hackbusch W (1985) Multi-grid methods and applications, vol 4. Springer, Berlin
- Haubner J, Siebenborn M, Ulbrich M (2021) A continuous perspective on shape optimization via domain transformations. *SIAM J Sci Comput* 43(3):A1997–A2018. <https://doi.org/10.1137/20m1332050>
- Hinze M, Pinnau R, Ulbrich M, Ulbrich S (2009) Optimization with PDE constraints. *Mathematical modelling: theory and applications*, vol 23. Springer, Berlin
- Iglesias JA, Sturm K, Wechsung F (2018) Two-dimensional shape optimization with nearly conformal transformations. *SIAM J Sci Comput* 40(6):A3807–A3830
- Jameson A (2003) Aerodynamic shape optimization using the adjoint method. *Lectures at the Von Karman Institute, Brussels*
- Karypis G, Schloegel K, Kumar V (2013) Parmetis, parallel graph partitioning and sparse matrix ordering library. Available at <http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview>
- Luft D, Schulz V (2021a) Pre-shape calculus: foundations and application to mesh quality optimization
- Luft D, Schulz V (2021b) Simultaneous shape and mesh quality optimization using pre-shape calculus
- Mannel F, Rund A (2020) A hybrid semismooth quasi-newton method for nonsmooth optimal control with PDEs. *Optim Eng*. <https://doi.org/10.1007/s11081-020-09523-w>
- Mohammadi B, Pironneau O (2010) Applied shape optimization for fluids. Oxford University Press, Oxford
- Murat F, Simon J (1976) Etude de problèmes d'optimal design. In: Cea J (ed) *Optimization techniques modeling and optimization in the service of man part 2: proceedings, 7th IFIP conference Nice, September 8–12, 1975*. Springer, Berlin, Heidelberg, pp 54–62
- Müller PM, Kühl N, Siebenborn M, Deckelnick K, Hinze M, Rung T (2021) A novel p-harmonic descent approach applied to fluid dynamic shape optimization. *Struct Multidiscip Optim* 64(6), 3489–3503
- Nägel A, Schulz V, Siebenborn M, Wittum G (2015) Scalable shape optimization methods for structured inverse modeling in 3D diffusive processes. *Comput Vis Sci* 17(2):79–88. <https://doi.org/10.1007/s00791-015-0248-9>
- Onyshkevych S, Siebenborn M (2021) Mesh quality preserving shape optimization using nonlinear extension operators. *J Optim Theory Appl* 165(5):291–316. <https://doi.org/10.1007/s10957-021-01837-8>
- Pinzon J, Siebenborn M (2021) Fluidoptim. Available at <http://www.github.com/multigriddshapeopt>
- Pinzon J, Siebenborn M, Vogel A (2020) Parallel 3d shape optimization for cellular composites on large distributed-memory clusters. *J Adv Simul Sci Eng* 7(1):117–135. <https://doi.org/10.15748/jasse.7.117>
- Reiter S, Vogel A, Heppner I, Rupp M, Wittum G (2013) A massively parallel geometric multigrid solver on hierarchically distributed grids. *Comp Vis Sci* 16(4):151–164

- Schmidt S, Ilic C, Schulz V, Gauger NR (2013) Three-dimensional large-scale aerodynamic shape optimization based on shape calculus. *AIAA Journal* 51(11):2615–2627
- Schulz V, Siebenborn M (2016) Computational comparison of surface metrics for PDE constrained shape optimization. *Comput Methods Appl Math* 16(3):485–496. <https://doi.org/10.1515/cmam-2016-0009>
- Schulz V, Siebenborn M, Welker K (2016) Efficient PDE constrained shape optimization based on Steklov-Poincaré -type metrics. *SIAM J Optim* 26(4):2800–2819. <https://doi.org/10.1137/15M1029369>
- Siebenborn M, Vogel A (2021) A shape optimization algorithm for cellular composites. *PINT Comput Vis Sci*. Available at arxiv.org/1904.03860
- Siebenborn M, Welker K (2017) Algorithmic aspects of multigrid methods for optimization in shape spaces. *SIAM J Sci Comput* 39(6):B1156–B1177
- Sokolowski J, Zolesio JP (2012) Introduction to shape optimization: shape sensitivity analysis, vol 16. Springer, Berlin
- Vogel A, Reiter S, Rupp M, Nägel A, Wittum G (2013) UG 4: a novel flexible software system for simulating PDE based models on high performance computers. *Comput Vis Sci* 16(4):165–179
- Wilke DN, Kok S, Groenwold AA (2005) A quadratically convergent unstructured remeshing strategy for shape optimization. *Int J Numer Methods Eng* 65(1):1–17. <https://doi.org/10.1002/nme.1430>
- Wittum G (1989) Multi-grid methods for stokes and Navier–Stokes equations. *Numer Math* 54:543–563. <https://doi.org/10.1007/BF01396361>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.