SHAPE OPTIMIZATION USING THE FINITE ELEMENT METHOD ON MULTIPLE MESHES WITH NITSCHE COUPLING

JØRGEN S. DOKKEN†, SIMON W. FUNKE †, AUGUST JOHANSSON †, AND STEPHAN SCHMIDT ‡

Abstract. An important step in shape optimization with partial differential equation constraints is to adapt the geometry during each optimization iteration. Common strategies are to employ mesh-deformation or re-meshing, where one or the other typically lacks robustness or is computationally expensive. This paper proposes a different approach, in which the computational domain is represented by multiple, independent meshes. A Nitsche based finite element method is used to weakly enforce continuity over the non-matching mesh interfaces. The optimization is performed using an iterative gradient method, in which the shape-sensitivities are obtained by employing the Hadamard formulas and the adjoint approach. An optimize-then-discretize approach is chosen due to its independence of the FEM framework. Since the individual meshes may be moved freely, re-meshing or mesh deformations can be entirely avoided in cases where the geometry changes consist of rigid motions or scaling. By this free movement, we obtain robust and computational cheap mesh adaptation for optimization problems even for large domain changes. For general geometry changes, the method can be combined with mesh-deformation or re-meshing techniques to reduce the amount of deformation required. We demonstrate the capabilities of the method on several examples, including the optimal placement of heat emitting wires in a cable to minimize the chance of overheating, the drag minimization in Stokes flow, and the orientation of 25 objects in a Stokes flow.

Key words. Shape Optimization, Finite Element Methods, MultiMesh FEM.

AMS subject classifications. 35Q93, 49Q10, 65M85, 65N30, 68N99.

1. Introduction. During the last two decades, there has been a transition from simulation to coupling of optimization and simulation [41]. Of particular industrial relevance are shape optimization problems, which aim to optimize the shape of an object subject to physical constraints, typically described by partial differential equations (PDEs). Examples of industrial problems that have been modeled are the drag minimization of airplanes and cars [26, 28, 32], the shape optimization of acoustic horns [36], and the optimal design of current carrying multi-cables [16]. The success of these applications is driven by efficient optimization algorithms and fast methods for solving PDEs. More specifically, gradient-based optimization methods have shown to converge quickly and often independent of the number of design variables. The required shape gradients are derived through shape calculus and the adjoint PDE [12, 35, 37]. The Finite Element Method (FEM) is an efficient and flexible method for solving a wide range of PDEs. In the last decades, this method has gained popularity in both the scientific and industrial environment due to its mathematical foundation and geometrical flexibility.

A critical part in shape optimization algorithms is handling of geometry changes during each optimization iteration. For FEM based models this means that the computational mesh must be updated to a new target geometry at low cost while maintaining a high mesh quality. Two commonly used mesh updating strategies are mesh deformation and re-meshing. Mesh deformation methods often involve the solution of an auxiliary PDE. However, the mesh quality may degrade or even degenerate for
large deformations. Several deformation schemes have therefore been proposed to handle large deformations \cite{34, 40} of the expense of a high computational cost. In contrast, a re-meshing strategy produces meshes that are guaranteed to be regular for any geometrical change. However, drawbacks are that the geometry must be reconstructed from the mesh to allow for re-meshing of the boundary elements, and the high computational cost of the meshing algorithms \cite{9}.

To overcome these limitations, we propose a shape optimization algorithm based on the idea to represent the domain by multiple, non-matching meshes, as illustrated in Figure 1. Because our method is highly embedded in the finite element setting, we call it MultiMesh as opposed to existing approaches like Chimera and Overset methods. Each mesh can be freely rotated, scaled or translated at a low computational cost without impacting the mesh quality. In an optimization setting, this means that mesh updates can be completely eliminated in cases where the goal is to optimally rotate, scale or translate of objects within a larger geometry. Further, as we will show in the numerical examples, the MultiMesh approach is beneficial for general shape optimization. Applying mesh deformation or re-meshing to a MultiMesh is computationally cheaper and more robust than the traditional single mesh approach, since it is only applied on submeshes. To the best of the authors’ knowledge, this is the first instance of a FEM with multiple overlapping meshes in the setting of shape optimization.

![Fig. 1. A comparison of a moving object described with a standard mesh and with multiple meshes. In (a), the mesh is deformed with an Eikonal convection equation, combined with a centroidal Voronoi tessellation (CVT) smoothing \cite{34}. The mesh quality, quantified by the minimum radius ratio decreases from 0.75 to 6 \cdot 10^{-4}, and the mesh is degenerated. In (b), a mesh describing the ball is introduced, and can be translated independent of the background mesh. Here the minimum radius ratio is constant at 0.72.](image)

The use of multiple meshes dates back to solving the problem of structure mesh generation for finite difference or structured finite volume schemes \cite{6, 17, 39, 44}. These many-mesh techniques (also known as Chimera or Overset techniques) overcome several limitations of structured grids, such as multiple holes and moving domains, making them particularly popular for aerodynamic applications \cite{38}. These schemes have also been used in an optimization setting, with similar data-transfer for the adjoint equation \cite{23}.

A recent method for non-matching meshes for FEM is the Cut Finite Element Method (CutFEM) \cite{10}. This method uses a Nitsche based formulation to weakly enforce boundary conditions at the interface. CutFEM has been used for a wide range
of shape and topology optimization problems, such as acoustics [8], elasticity [5, 11] and incompressible flow [43]. The MultiMesh FEM [19] is a generalization of the CutFEM, where the computational domain is described by arbitrary many overlapping non-matching meshes, coupled with Nitsche’s method. The MultiMesh FEM has been explored for the Poisson and Stokes-equations [15, 21, 22]. Other methods used for shape optimization of complex computational domains are available, see for example [27, 30, 42] and the references therein.

The MultiMesh FEM introduces several interesting aspects when applied to a shape optimization problem. When creating a solution algorithm for the optimization problem, a choice has to be made, namely, should one use the first optimize, then discretize approach or the first discretize, then optimize approach. An initial analysis of these approaches has been investigated with respect to shape optimization problems [7]. However, there is no general recipe for which method is to be preferred [18]. Due to generality and independence of the FEM framework, we have chosen the optimize then discretize approach. However, the authors plan to investigate the effects of a discretize then optimize strategy in a later publication. Numerical examples show that the numerical inconsistency in the shape gradient is insignificant for fine meshes.

The remainder of this paper is organized as follows. Section 2 introduces the mathematical notation and presents the new algorithm for solving shape optimization on multiple domains. Section 3 gives a brief introduction solving PDEs on multiple meshes with the MultiMesh FEM. In Section 4, we then derive shape derivatives using shape calculus, and the associated adjoint equations. Section 5 discusses the optimization step and the mesh updating strategies. Thereafter, we present several numerical examples in Section 6 and compare the new approach to a traditional FEM when feasible. Finally, we summarize and draw conclusions in Section 7.

2. Algorithm for solving shape optimization on multiple overlapping domains. In this section, we present the algorithm for solving PDE constrained shape optimization problems using the MultiMesh Finite Element Method (FEM).

In this paper, we consider shape optimization problems of the form

\[
\min_{u,\Omega} J(u, \Omega) \\
\text{subject to} \\
E(u, \Omega) = 0
\]

(2.1)

where \(J(u, \Omega) \in \mathbb{R}\) is an objective functional, \(E(u, \Omega) = 0\) is a PDE with solution \(u\) defined over the polynomial domain \(\Omega \subset \mathbb{R}^n, n = 1, 2, 3\).

A related and common case is where the domain \(\Omega\) is parameterized and the goal is to optimize the design parameters. The following example illustrates this.

Example: Consider the problem of minimizing the squared \(L_2\)-norm of the solution of the Poisson equation. The domain contains an obstacle, which may be rotated freely around a point \(p\). The objective is to determine the optimal rotation angle \(\theta\) of the obstacle. The optimization problem takes the form

\[
\min_{T,\theta} J(T, \theta) = \min_{T,\theta} \int_{\Omega(\theta)} T^2 \, dx,
\]

(2.3)
subject to

\[ -\Delta T = f \quad \text{in } \Omega, \]
\[ T = g \quad \text{on } \partial \Omega, \]
\[ g = \begin{cases} 1 & \text{on } \Gamma \subset \partial \Omega, \\ 0 & \text{on } \partial \Omega \setminus \Gamma. \end{cases} \]

(2.4)

Here, the domain \( \Omega \) consists of a rectangle with an elliptic obstacle parameterized by \( \theta \). Further, the boundary of \( \Omega \) including the boundary of the obstacle is denoted by \( \partial \Omega \). The boundary of the obstacle is denoted by \( \Gamma \subset \partial \Omega \) and \( f(x, y) = x \sin(x) \cos(y) \) is the source function. The setup is visualized in Figure 2.

Fig. 2. The setup of the example (2.3) and (2.4). The obstacle, marked in gray, is rotated 45° around the point \( p \).

After differentiation, we discretize the shape optimization problem (2.1–2.2) on multiple domains. For that, the domain \( \Omega \) is divided into multiple, possibly overlapping subdomains \( \hat{\Omega}_i, i = 0, \ldots, N \). Further, we restrict the PDE operator \( E \) and the state variable \( u \) onto these subdomains. We denote these restrictions as \( E_{\hat{\Omega}_i} \) and \( u_i \). The optimization problem (2.1-2.2) can now be reformulated to:

\[ \min_{u, \hat{\Omega}_0, \hat{\Omega}_1, \ldots, \hat{\Omega}_N} J(u, \hat{\Omega}_0, \ldots, \hat{\Omega}_N) \]

subject to

\[ E_{\hat{\Omega}_i}(u_i, \hat{\Omega}_i) = 0, \quad i = 0, \ldots, N, \]
\[ E_{\Lambda_j}(u_1, \ldots, u_j, \hat{\Omega}_1, \ldots, \hat{\Omega}_j) = 0, \quad j = 1, \ldots, N, \]

(2.6)

where \( E_{\Lambda_j} \) are interface conditions that arise from the MultiMesh FEM (see Section 3), which ensures equivalency of (2.2) and (2.6).

Example (cont.): A domain composition of the example problem is visualized in Figure 3.

Formulation (2.5–2.6) is very convenient for discretizing the shape derivative of (2.3–2.4), since each domain can be updated separately.
SHAPE OPTIMIZATION USING MULTIMESH FEM WITH NITSCHE COUPLING

5

Fig. 3. Subdomains of the multiple domain formulation for example (2.3–2.4). The domain $\hat{\Omega}_1$ and the obstacle, marked in gray, can be rotated around $p$.

The formulation (2.5–2.6) results in the following solution algorithm:

Algorithm 2.1 Algorithm for shape optimization with multiple domains.

Set iteration counter $k = 0$
Choose subdomains $\hat{\Omega}^k_i, i = 0, \ldots, N$
while not converged do
Solve the state equations (2.6) on $\bigcup_{i=0}^{N} \hat{\Omega}^k_i$ with MultiMesh FEM (Section 3)
Compute the shape sensitivities of functional (2.5) (Section 4)
Perform optimization step to obtain subdomains $\hat{\Omega}^{k+1}_i, i = 0, \ldots, N$ (Section 5)
Set $k \leftarrow k + 1$
end while
return Optimized domain $\bigcup_{i=0}^{N} \hat{\Omega}^k_i$

3. The MultiMesh Finite Element Method. In this section, we will discuss how to solve the state equation (2.2) with a domain consisting of multiple overlapping sub-domains using the MultiMesh FEM, for which we recall some notation from [19].

3.1. Domains and Meshes. In Section 2, we introduced a composition of $\Omega$, such that $\Omega \subseteq \bigcup_{i=0}^{N} \hat{\Omega}_i$, where $\hat{\Omega}_i$ is defined as the predomain. If a point $x \in \Omega$ can be found in multiple predomains, we associate it with the highest index $i$. Thus, if interpreted visually, the predomain with the higher index appears to be on top of the predomain with the lower index. Since the predomains will overlap, we define the visible part of each predomain as $\hat{\Omega}_i := \hat{\Omega}_i \setminus \bigcup_{j=i+1}^{N} \hat{\Omega}_j, i = 0, \ldots, N$. Note that $\hat{\Omega}_N = \hat{\Omega}_N$. The visible boundary of each predomain $\hat{\Omega}_i$ is denoted $\Lambda_i := \partial \hat{\Omega}_i \setminus \bigcup_{j=i+1}^{N} \hat{\Omega}_j, i = 0, \ldots, N$.

We define a premesh $\hat{K}_{h,i}$ as the mesh of the predomain $\hat{\Omega}_i$, and denote its maximum cell diameter $h_i$. The elements of $\hat{K}_{h,i}$ can be divided into the following three distinct categories: uncut, cut and covered elements. Uncut elements are the fully visible elements, cut elements are the partially visible elements, and covered elements are the hidden elements. By manually changing the status of elements to covered, topological changes such as holes may be modeled. This is illustrated in the next example.

The $i$-th active mesh $K_{h,i}$ consists of all cut and uncut elements of $\hat{K}_{h,i}$. We define the cut domain $\Omega^\text{cut}_i$ as the union of all cut elements. Note that $\Omega^\text{cut}_N = \emptyset$. The $i$-th overlap is defined as $\mathcal{O}_i = \Omega^\text{cut}_i \setminus \hat{\Omega}_i, i = 0, \ldots, N - 1$. This is the hidden part of
Example (cont.): The predomains $\hat{\Omega}_0$ and $\hat{\Omega}_1$ are shown in Figure 3. In Figure 4 the visible part of each domain, that is, $\Omega_0$ and $\Omega_1$, is illustrated. In Figure 5a), the premeshes $\hat{K}_{h,0}$ and $\hat{K}_{h,1}$ are illustrated in black and red. Figure 5a) also shows the cut, uncut and covered elements. Note that all element in $\hat{K}_{h,1} = K_{h,1}$ are uncut. In Figure 4, the visible part of each domain, that is, $\Omega_0$ and $\Omega_1$, is illustrated. In Figure 5a), a hole has been introduced in the domain by setting all elements in $\hat{K}_{h,0}$ that are cut or covered by the obstacle on $\hat{\Omega}_0$ to being covered. This creates the effect of a hole in $\hat{\Omega}_0$, since the covered elements will be ignored in the weak formulation. The boundary of the obstacle now becomes a physical boundary, $\Gamma := \partial \Omega_1 \setminus \Lambda_1$. This will be discussed in the next section. The strong form of the state equations, (2.4) can be written as

\[ E_{\Omega_0}(T_0, \Omega_0) = -\Delta T_0 - f = 0 \quad \text{in} \; \Omega_0, \quad T_0 = g, \quad \text{on} \; \partial \Omega_0, \]

\[ E_{\Omega_1}(T_1, \Omega_1) = -\Delta T_1 - f = 0 \quad \text{in} \; \Omega_1, \quad T_1 = g, \quad \text{on} \; \Gamma, \]

\[ E_{\Lambda_1}(T_0, T_1, \Omega_0, \Omega_1) = \begin{cases} \|T\| = 0 & \text{on} \; \Lambda_1, \\ n_1 \cdot \nabla T = 0 & \text{on} \; \Lambda_1, \end{cases} \]

where $\cdot$ is the vector dot product, $\|\psi\| = \psi_1 - \psi_0$ denotes the jump. The normal vector $n_1$ is defined to be pointing outwards of the domain $\hat{\Omega}_1$. The two interface conditions on $\Lambda_1$ ensure sufficient smoothness of the solution $T$ across the $\Lambda_1$.

3.2. Function Spaces and the Finite Element Method. For the weak formulation of (3.1) and (3.2) the interface conditions are enforced weakly using a Nitsche method [29]. The method contains interior penalty terms, very similar to a discontinuous Galerkin method [4], as well as additional stabilization to obtain a stable method. The proposed method is symmetric, stable and yields optimal convergence rates and optimal condition numbers, also in the case of small overlaps [19]. Since the interface is not aligned with the meshes, custom quadrature rules are needed to perform the volume and interface integrals that appear in the formulation. See [20] for details. Let $V_{h,i}$, $i = 0, \ldots, N$, denote the finite element space on the active mesh $\hat{K}_{h,i}$ consisting of continuous piece-wise Lagrange polynomials. We define $V_h := \bigoplus_{i=0}^N V_{h,i}$.

Example (cont.): Let $V_h$ be as described above using $N = 1$, and let $V_h^g$ denote the corresponding function space that satisfy the boundary condition. The MultiMesh finite element formulation of the aforementioned example is the following:
SHAPE OPTIMIZATION USING MULTIMESH FEM WITH NITSCHE COUPLING

Find $T = (T_0, T_1) \in V_h^g$ such that

$$a(T, v) + a_{IP}(T, v) + a_O(T, v) = l(v) \quad \forall v = (v_0, v_1) \in V_h^0,$$

with

$$a(T, v) := \sum_{i=0}^{1} \int_{\Omega_i} \nabla T_i \cdot \nabla v_i \, dx,$$

$$l(v) := \sum_{i=0}^{1} \int_{\Omega_i} f v_i \, dx.$$

The symmetric interior penalty terms are

$$a_{IP}(T, v) := \int_{\Lambda_1} \langle n_1 \cdot \nabla T \rangle \|v\| - \langle T \rangle \langle n_1 \cdot \nabla v \rangle + \frac{\beta_0}{h} \|T\| \|v\| \, dS,$$

where $\langle n_1 \cdot \nabla \psi \rangle = \frac{1}{2} n_1 \cdot (\nabla \psi_1 + \nabla \psi_0)$ is the average, $\beta_0$ is a sufficiently large penalty parameter and $h = h_0 + h_1$. The overlap stability term is

$$a_O(T, v) := \int_{\tilde{\Omega}_1} \beta_1 \|\nabla T\| : \|\nabla v\| \, dx,$$

where $\beta_1$ is added for controlling the conditioning of the arising linear system. If not otherwise stated, $\beta_0 = \beta_1 = 4$ is used.

4. Shape Calculus. The goal of the section is to derive the (shape) derivative of the objective functional (2.5) with respect to the domain $\Omega$. In principal, the shape derivatives can be derived before or after the MultiMesh FEM discretization. We perform the derivation before the discretization (i.e. from (2.1) and (2.2)), which has the benefits as discussed before, namely independence of the software and multi-mesh formulation to be used. The downside is that we introduce a discrete inconsistency in the shape gradient. This inconsistency is only affecting performance when employing coarse meshes. On finer meshes, the inconsistency does not affect performance, as shown later in this section. The chosen approach is visualized in Figure 6.

We assume that the state (2.2) yields a unique solution $u$ for any given domain $\Omega$. Then we define the reduced functional as $\hat{J}(\Omega) = J(u(\Omega), \Omega)$. We define a perturbed domain $\Omega(\epsilon)$ as

$$\Omega(\epsilon)[s] := L_\epsilon[s](\Omega) = \{ L_\epsilon[s](x) : x \in \Omega \},$$

5. Numerical Results. The numerical results are shown in this section. The results are compared to the analytical solution. The numerical solution is shown to be accurate.

Fig. 5. (a) Visualization of the premeshes $\hat{K}_0$ (black) and $\hat{K}_1$ (red). The uncut, cut and covered elements of $\hat{K}_0$ is shown. (b) The element types after introducing a hole in the domain.
where \( L_\epsilon[s](x) := x(\epsilon) := x + \epsilon s(x) \), \( s(x) : \mathbb{R}^n \to \mathbb{R}^n, \epsilon > 0 \). The shape derivative is then defined as

\[
(4.2) \quad d \hat{J}(\Omega)[s] := \lim_{\epsilon \to 0^+} \frac{\hat{J}(\Omega(\epsilon)) - \hat{J}(\Omega)}{\epsilon}.
\]

The solution of the PDE on the perturbed domain \( \Omega(\epsilon) \) is denoted \( u_\epsilon \). The material derivative and local derivative of \( u_\epsilon \) are defined as

\[
(4.3) \quad du[s] := \lim_{\epsilon \to 0^+} \frac{u(\epsilon,x(\epsilon)) - u(0,x(0))}{\epsilon}, \quad u'[s] := du[s] - Du \cdot s,
\]

where \( Du \) is the Jacobian. Using Hadamard’s formula, we find the total shape derivative of the functional.

**4.1. Hadamard’s formulas.** We consider the cases where the functional is a volume integral or surface integral. The surface formulation of the Hadamard formulas is used, as the volume formulation would include expressions containing the overlap and cut interface. The authors plan to investigate the effects of these additional terms in a later publication.

**Theorem 4.1 (Hadamard Formula for Volume Objective Functions).** For a general volume objective function \( k : \Omega \to \mathbb{R} \)

\[
(4.4) \quad J(\Omega) = \int_\Omega k \, dx,
\]

the shape derivative is given by

\[
(4.5) \quad dJ(\Omega)[s] = \int_\Gamma s \cdot nk \, dS + \int_\Omega k'[s] \, dx.
\]

**Proof.** To facilitate the derivative with respect to the perturbed domain, the
integral is transported to the unperturbed domain, as shown below.

\[
\begin{align*}
\frac{dJ(\Omega)[s]}{ds} &= \int_\Omega k(\cdot, \epsilon) \, dx = \int_\Omega \left( \frac{d}{d\epsilon} \right)_{\epsilon=0} \left( k(L_\epsilon(\cdot, \epsilon)|\det DL_\epsilon(\cdot)|) \right) \, dx \\
&= \int_\Omega \left( \frac{d}{d\epsilon} \right)_{\epsilon=0} \left( k(\cdot, \epsilon) \right) \, dx \\
&= \int_\Gamma s \cdot n \, k \, dS + \int_\Omega k'[\cdot] \, dx.
\end{align*}
\]

More details can be found in [12].

**Theorem 4.2 (Hadamard Formula for Surface Objectives).** For a general surface objective function \( h : T(\Gamma) \rightarrow \mathbb{R} \), which is dependent of the shape and for which \( \frac{\partial h}{\partial n} \) exists, the shape derivative for the surface objective

\[
J(\Omega) = \int_\Gamma h \, dS
\]

is given by

\[
\frac{dJ(\Omega)[s]}{ds} = \int_\Gamma s \cdot n \left( \frac{\partial h}{\partial n} + \gamma h \right) \, dS + \int_\Gamma h'[\cdot] \, dS,
\]

where \( \gamma = \text{div}_n \) is the tangential divergence of the normal, i.e. the additive mean curvature of \( \Gamma \).

This proof is following the same strategy as Theorem 4.1. Therefore the proof is omitted, and given in [12].

In our case, \( h'[\cdot] \) and \( k'[\cdot] \) are the local derivatives of the state variable \( u \) with respect to the design parameters. When discretized, this is a dense matrix which is prohibited to compute. Instead, we use the adjoint approach to avoid explicit computations of these terms.

**4.2. The Adjoint Approach.** We use the Lagrangian approach to obtain the shape sensitivity and the adjoint equation. We start by defining the Lagrangian \( \mathcal{L} \) of problem (2.1–2.2), assuming that \( J \) is a volume integral

\[
\mathcal{L}(u, \Omega, \lambda) = \int_\Omega j \, dx + (\lambda, E(u, \Omega))_{\Omega},
\]

where \( \lambda \) is the Lagrange multiplier. Technically, the scalar product used to define the Lagrangian should not depend on the design parameter \( \Omega \). However, as shown above in the proof of Theorem 4.1 the shape gradient is expressed on a reference domain.

The directional derivative of the Lagrangian is

\[
\frac{d\mathcal{L}[s]}{ds} = \int_\Gamma s \cdot n (j + \lambda E) \, dS + \int_\Omega j'[\cdot] + \lambda E'[\cdot] \, dx + \int_\Omega \lambda'[\cdot] E \, dx.
\]
The necessary optimality condition states that the directional derivative of the Lagrangian vanishes for all \( s \). This yields the following conditions in variational form.

\[
\int_{\Gamma} s \cdot n (j + \lambda E) \, dS = 0 \quad \forall s, \quad \text{(Design Eq.)}
\]

\[
\int_{\Omega} \partial j \frac{u'}{u} + \lambda \frac{\partial E}{\partial u} u' \, dx = 0 \quad \forall u' \quad \text{(Adjoint Eq.),}
\]

\[
\int_{\Omega} \lambda' E \, dx = 0 \quad \forall \lambda' \quad \text{(State Eq.).}
\]

Example (cont.): For the example problem (2.3–2.4), the Lagrangian is

\[
L(T, \Omega, (\lambda^\Omega, \lambda^\partial)) = \int_{\Omega} T^2 \, dx + \int_{\Omega} \lambda^\Omega (\Delta T - f) \, dx + \int_{\partial\Omega} \lambda^\partial (T - g) \, dS
\]

The adjoint equation (4.12) is

\[
\int_{\Omega} 2TT' \, dx + \int_{\Omega} -\lambda^\Omega \Delta T' \, dx + \int_{\partial\Omega} \lambda^\partial T' \, dS = 0 \quad \forall T'.
\]

Integrating by parts twice, assuming sufficient regularity of \( \lambda \), and interpreting the result as the strong formulation yields

\[
-\Delta \lambda^\Omega = -2T \quad \text{in } \Omega,
\]

\[
\lambda^\Omega = 0 \quad \text{on } \partial\Omega,
\]

\[
\lambda^\partial = -\frac{\partial \lambda^\Omega}{\partial n} \quad \text{on } \partial\Omega.
\]

We observe that the adjoint equation (4.16) is a Poisson problem with a homogeneous Dirichlet condition on \( \partial\Omega \). Using the notation of (3.3), we obtain the weak formulation on two meshes of the adjoint equation as: Find \( \lambda = (\lambda_0, \lambda_1) \in (V^\partial_0, V^\partial_1) = V \) such that:

\[
a(\lambda, v) + a_{IP}(\lambda, v) + a_G(\lambda, v) = -2 \sum_{i=0} T_i v_i \, dx, \quad \forall v = (v_0, v_1) \in V.
\]

The design equation (4.11) is

\[
\int_{\Gamma} s \cdot n \left( T^2 + \lambda^\Omega (\Delta T - f) + \frac{\partial (\lambda^\partial (T - g))}{\partial n} + \kappa \lambda^\partial (T - g) \right) \, dS = 0 \quad \forall s.
\]

Assuming that state (2.4) and adjoint (4.16) equations are satisfied, the left hand side of the design equation corresponds to the gradient of the reduced functional, i.e.

\[
d\hat{J}(\Omega)[s] = \int_{\Gamma} s \cdot n \left( T^2 - \frac{\partial \lambda^\Omega}{\partial n} \frac{\partial (T - g)}{\partial n} \right) \, dS.
\]
5. Optimization Algorithm and Mesh Deformation. We use the steepest
descent method as the optimization algorithm. Combined with a line search, the
steepest descent method guarantees a decrease in the goal functional by iteratively
moving in the opposite direction as the gradient of the functional. More precisely,
for a functional $\hat{J}$, with design-parameter $\Omega^k$ at iterate $k$, the $k$-th iteration of the
steepest descent method is:

$$\Omega^{k+1} = \Omega^k(\xi)[d]$$

where $\xi > 0$ is the step-length decided by e.g. an Armijo linesearch [3], and $d : \Omega^k \to \mathbb{R}^n$ is the Riesz representer of $-d\hat{J}$. It is convenient to incorporate the mesh deforma-
tion scheme at this point, as it ensures continuity into the interior and smoothness
of the boundary. This can be achieved for instance by choosing a Riesz representer in
a scaled $H^1$-norm. This results in the following variational problem: find $d$ such that

$$\int_{\Omega^k} \alpha \nabla d \cdot \nabla s + d \cdot s \, dx = -d\hat{J}[s] \quad \forall s,$$

where $\alpha \geq 0$ can be thought of as a smoothing parameter.

By choosing appropriate test and trial function spaces for (5.2), we can control
which type of shape deformations are allowed. For instance, if the design variable
is the position of an obstacle, it is natural to create a submesh that contains the obstacle,
and to choose the test and trial functions to constant unit vectors restricted to that
submesh. The consequence is that the Riesz-representer $d$ is a spatially constant
function on the submesh, and the submesh is translated in its entirety, see Figure 1b).
This is in contrast to a traditional one mesh approach, where the test and trial function
spaces are linear functions per element. This results in the compression effect, see
Figure 1a), and can lead to invalid meshes for larger deformations. Similarly, if the
design variable is the rotation of an obstacle contained in a submesh $\hat{\Omega}^k_i \subset \mathbb{R}^2$, the
test and trial space is spanned by a single function describing the rotation velocity of
the submesh:

$$s(p) = \begin{cases} 
(p_y + c_y, p_x - c_x), & \text{if } p \in \hat{\Omega}^k_i, \\
(0, 0), & \text{else},
\end{cases}$$

where $c = (c_x, c_y)$ is the center of rotation.

In the general case, where the design variables are the node positions on a bound-
ary $\Gamma$ on a submesh $\hat{\Omega}^k_i$, a natural test and trial space for (5.2) is the finite element
space spanned by continuous, piece-wise linear functions on $\hat{\Omega}^k_i$. This approach works
well for small deformations, but yields degenerated meshes for larger deformations. A
more robust, but computationally more expensive approach is to additionally advect
the negative gradient from $\Gamma$ to the other boundaries

$$\int_{\hat{\Omega}^k_i} \alpha \nabla d \cdot \nabla s + d \cdot \nabla \epsilon \cdot \nabla s \, dx = 0 \quad \forall s \text{ on } \hat{\Omega}^k_i,$$

$$d = ng(x) \quad \text{on } \Gamma,$$

where the right hand side stems from the integrand of the shape derivative, i.e.
\[ d \hat{J}[t] = \int_\Gamma t \cdot n g(x) \, d\Gamma, \]
and \( \epsilon \) is the solution of a smoothed Eikonal equation:

\[
-\alpha_1 \Delta \epsilon + \|\nabla \epsilon\|_2^2 = 1 \quad \text{in } \hat{\Omega}_k,
\]

\[
\epsilon = 0 \quad \text{on } \Gamma,
\]

with smoothing parameter \( \alpha_0, \alpha_1 \geq 0 \). The results in this paper have been produced with \( \alpha_0 = 10^{-3}, \alpha_1 = 25 \) unless otherwise stated. Note that the outer boundaries of the submesh \( \partial \hat{\Omega}_k \setminus \Gamma \) are free to move. Further note that the deformation equations only have to be solved on the submesh, while the background domain is kept stationary. Therefore, the degrees of freedom in the deformation equation are significantly reduced compared to the traditional one mesh approach.

The other alternative for updating the computational domain is to move the boundary of the original mesh, and use a re-mesh algorithm based on the new boundary. This approach is not employed in this article, but the authors note that by employing multiple meshes, smaller meshes can be re-meshed to make large changes in the geometry.

Example (cont.): We illustrate the advantage of the multiple domain approach with the example problem (2.3) and (2.4). The optimal rotation of the obstacle is shown in Figure 7 b), where the optimization algorithm converged after 5 iterations. Since the top-mesh \( \hat{K}_{h,1} \) has been rotated as an entity, the mesh-quality is fully preserved. In Figure 7 a), we verify the solution by plotting the functional for all different orientations of the obstacle.

![Figure 7](image)

Fig. 7. (a) The functional \( \hat{J} \) as a function of the rotation angle \( \theta \). (b) The optimal orientation of the obstacle, with angle \( \theta = 296.6 \) where the domain for the initial domain is \( \theta = 0.0 \). The optimized orientation was achieved after 5 steepest-descent iterations, and a total of 12 functional evaluations, including those in the Armijo linesearch.

6. Numerical examples.

6.1. Implementation. The MultiMesh FEM is implemented in the finite element framework FEniCS [1, 24], and will be released in version 2018.1.0. All meshes used in this paper have been generated with GMSH, version 3.0.6 [14]. The complete code of the examples are published at Bitbucket and access can be granted by emailing the corresponding author.

6.2. Optimization of Current Carrying Multi-cables. In modern cars, the number of electronic devices have been increasing, especially in electric and hybrid
cars. This means that car manufacturers must design wires carrying current to the different devices as compactly as possible. An example of such a multi-cable is shown in Figure 8. This motivates optimizing the design of such multi-cable to minimize the heat inside, see [16] and the references therein. As design variables, the authors chose the position of each internal cable of the multi-cable. Each optimization iteration results in new cable positions and a re-meshing strategy was used to update the mesh. Since the internal cables are translated within the multi-cable, the re-meshing step can be avoided if we apply Algorithm 2.1.

To demonstrate this, we consider a simplified multi-cable optimization problem:

\begin{equation}
\min_{\Omega,T} J(\Omega,T) = \int_{\Omega} \frac{1}{q} |T|^q \, dx, \quad q > 1,
\end{equation}

subject to

\begin{align*}
- \nabla \cdot (\lambda \nabla T) - cT &= f \text{ in } \Omega, \\
\lambda \frac{\partial T}{\partial n} + (T - T_{ex}) &= 0 \text{ on } \partial\Omega, \\
[T]_{\Gamma_i^{j/e}} &= 0 \text{ on } \Gamma_i^j \cup \Gamma_i^e, j = 1, \ldots, N,
\end{align*}

where \( \Omega = \Omega_{fill} \cup \Omega_{insulation} \cup \Omega_{metal} \) describes a 2D slice through the multi-cable with \( N \) internal cables, as specified in Figure 9 a). The operation \([\ ]_{\Gamma_i^{j/e}}\) denotes the jump over the interface \( \Gamma_i^j \) and \( \Gamma_i^e \). The state equation is a Poisson equation with an additional linear source term with temperature coefficient \( c = 0.04 \). This term describes the rise of electrical resistivity for increasing temperatures in conductive material. The external boundary condition is a Robin-condition, related to the air surrounding the cable, with temperature \( T_{ex} = 3.2 \). Furthermore, we set \( q = 3 \) to approximate the \( L^\infty \) norm, as done by [16]. A detailed derivation of these equations can be found in [25]. The source-term \( f \) and heat-conductivity \( \lambda \) are discontinuous, piece-wise constant functions with the following values:

|        | \( \Omega_{fill} \) | \( \Omega_{insulation} \) | \( \Omega_{metal} \) |
|--------|----------------|----------------|----------------|
| \( f \) | 0.0           | 0.0           | 50.0          |
| \( \lambda \) | 0.08       | 0.19         | 40.0          |
The state equation is solved with the MultiMesh FEM for arbitrarily many intersecting meshes, see Section 3 and [19]. There are different ways of creating the overlapping meshes for this problem. We chose to represent the domain by one mesh for the filling material, and \( N \) meshes for the inner cables, see Figure 9b). The meshes for the internal cables include a halo of filling material, which was chosen sufficiently large so that the heat conductivity \( \lambda \) is constant over the cells categorized as overlapped. This guarantees that the solution on the overlap area does not add non-physical contributions to the weak formulation.

We choose the position of each inner cable as a design variable, as described in Section 5. Since the background cable is fixed, there are additional constraints on the centroids. The distance of each cable to origin has to be bounded, such that the cables do not move outside the cable defined by \( \Gamma^{ex} \). To enforce these constraints, we used a projected Armijo Rule [18].

The adjoint equations are derived following subsection 4.2 and are:

\[
\begin{align*}
-\nabla \cdot (\lambda \nabla p) - cp &= -T|T|^q - 2 \quad \text{in } \Omega, \\
p &= p^{ex} \quad \text{on } \partial \Omega, \\
\lambda \frac{\partial p}{\partial n} + p^{ex} &= 0 \quad \text{on } \partial \Omega, \\
\left[ \lambda \frac{\partial p}{\partial n} \right]_{\Gamma_j^{i/e}} &= 0 \quad \text{on } \Gamma_j^{i} \cup \Gamma_j^{e}, j = 1, \ldots, N, \\
\left[ p \right]_{\Gamma_j^{i/e}} &= 0 \quad \text{on } \Gamma_j^{i} \cup \Gamma_j^{e}, j = 1, \ldots, N.
\end{align*}
\]

(6.3)

and the shape sensitivity is

\[
\frac{dJ(\Omega)}{ds} = \sum_{j=1}^N \int_{\Gamma_j^{i} \cup \Gamma_j^{e}} s \cdot n \left( \left\| -c T p - f p \right\|_{\Gamma_j^{i/e}} - \lambda \frac{\partial p^+}{\partial n} \left[ \frac{\partial T}{\partial n} \right]_{\Gamma_j^{i/e}} \right.
\]

\[
\left. + \left\| \lambda \nabla T p^+ \cdot \nabla T^+ \right\|_{\Gamma_j^{i/e}} \right) dS,
\]

(6.4)
SHAPE OPTIMIZATION USING MULTIMESH FEM WITH NITSCHE COUPLING

where the super-script + denotes the evaluation of a function from the fill side at \( \Gamma^j \), and evaluation at the insulation side of \( \Gamma^i \).

6.2.1. Results. Numerical results were performed on a multi-cable with radius 1.2, containing \( N \) internal cables with 0.2 radius not counting insulation and a 0.055 thick insulation.

The adjoint equation and shape sensitivity were verified by a Taylor test, where one inner cable without insulation was placed at \((0.03, 0.2)\). A MultiMesh consisting of a background mesh with 33802 elements and 15842 elements for the mesh of each of the internal cable was used. The convergence rates in the steepest descent direction are shown in Table 1. Similar convergence rates were obtained for other perturbations, indicating that the adjoint equation and the shape derivatives are correct. Further tests with different mesh refinements showed that the Taylor convergence rates decreases if the mesh is too coarse, due to the discrete inconsistency in the functional sensitivities, see Section 4.

Table 1

| \( \epsilon \) | \( R_0(\epsilon) \) | order | \( R_1(\epsilon) \) | order |
|----------------|-----------------|-------|-----------------|-------|
| 1.70e-04       | 2.404e+03       | -     | 1.089e+03       | -     |
| \( \epsilon/2 \) | 9.743e+02       | 1.30  | 3.168e+02       | 1.78  |
| \( \epsilon/4 \) | 4.112e+02       | 1.24  | 8.247e+01       | 1.94  |
| \( \epsilon/8 \) | 1.847e+02       | 1.15  | 2.055e+01       | 2.02  |
| \( \epsilon/16 \) | 8.697e+01       | 1.09  | 4.783e+00       | 2.09  |
| \( \epsilon/32 \) | 4.213e+01       | 1.05  | 1.034e+00       | 2.21  |

Next, the optimization algorithm was verified by optimizing the position of three identical cables. For this setup, it is known that the optimal positioning of the cables form an equilateral triangle [16]. The optimization loop was terminated when the relative functional reduction dropped below a set tolerance, i.e. \( \left| \frac{J(\Omega^{k+1}) - J(\Omega^k)}{J(\Omega^{k+1})} \right| < 10^{-6} \). The optimal configuration was found after 62 iterations, when the functional decreased from 3.2 \cdot 10^4 to 3.9 \cdot 10^3. The optimized angles between the cables were 58.7, 62.1, 59.3 degrees. The initial and optimized configurations are shown in Figure 10.

Furthermore, we compared the computational expense of the MultiMesh FEM with a traditional FEM approach. For this comparison, we considered a problem with one internal cable. We measured the run-time of one approximate iteration, consisting of assembling and solving the state and adjoint systems, and updating the mesh. For the MultiMesh-approach, the mesh update consists of translating the mesh coordinates of the inner cable and to recompute the new mesh intersections. For the traditional one mesh approach, the mesh update was performed through remeshing. The linear systems arising in both approaches were solved using the an LU solver. The timing results are shown in Table 2. The results show that the assembly of the MultiMesh-system is more time consuming than the traditional one mesh approach, primarily caused by the the additional stabilization terms. However, this additional expense is outweighed by a significant lower cost for the mesh update compared to re-meshing. Therefore, the estimated iteration cost for the MultiMesh-approach is a third of the traditional approach.

Finally, the optimization was performed on a problem with five internal cables of different sizes and with different insulation parameters. The parameters are listed...
in Table 3. The initial and optimized cable configurations are shown in Figure 11. The results show that the smallest cable, Cable 5, is placed far away from the other cables. This can be explained through the fact that this cable has the lowest insulation parameter $\lambda_{\text{iso}}$ and the largest heat source $f$. The optimal solution was found after 103 iterations, and the functional decreased from $1.1 \cdot 10^5$ to $9.8 \cdot 10^3$.

![Figure 10](image)

**Fig. 10.** (a) The cable configuration and temperature distribution the three cables before the optimization. (b) The cable configuration and temperature distribution after the optimization. The inner cables from an equilateral triangle.

**Table 2**

The minimum time each operation takes in seconds (5 measures). The remeshing of a single mesh has been done with GMSH 3.0.6 [14], where we have measured the time it takes to convert the traditional mesh geo-file to a msh file. Note that assembling the MultiMesh variational form takes more time than the traditional FEM system, due to the additional terms in the variational form. A typical iteration in an optimization algorithm (excluding line searches) includes two assembly and solves (state and adjoint) equation, one mesh update (re-meshing or translation), and a recomputation of intersections for the multimesh (build).

| Cells | Assembly | Solve | Mesh Update | Build | App. It. |
|-------|----------|-------|-------------|-------|----------|
| MultiMesh FEM | 47728 | 1.39e-01 | 2.10e-04 | 3.81e-02 | 5.30e-01 |
| Traditional FEM | 46178 | 8.44e-02 | 1.12e-01 | 1.12e+00 | - | 1.51e+00 |

**Table 3**

The setup for the 5 multi-cable optimization shown in Figure 11. The parameters $\lambda_{\text{fill}}, \lambda_{\text{metal}}$, are the same as in the other simulations.

| Cable 1 | Cable 2 | Cable 3 | Cable 4 | Cable 5 |
|---------|---------|---------|---------|---------|
| Init. Positions | $0, 0.45$ | $-0.4, -0.15$ | $0.2, -0.4$ | $0.5, 0.15$ | $0.7, -0.3$ |
| Opt. Positions | $-0.434, 0.777$ | $-0.903, 0.060$ | $-0.191, -0.9$ | $0.536, 0.784$ | $0.902, -0.384$ |
| $\lambda_{\text{iso}}$ | $0.255$ | $0.242$ | $0.218$ | $0.174$ | $0.172$ |
| $\lambda_{\text{metal}}$ | $0.2$ | $0.19$ | $0.171$ | $0.137$ | $0.096$ |
| $f$ | $0.19$ | $0.162$ | $0.133$ | $0.105$ | $0.076$ | $50$ | $60$ | $70$ | $80$ | $90$ |
SHAPE OPTIMIZATION USING MULTIMESH FEM WITH NITSCHE COUPLING

6.3. Shape Optimization of an Obstacle in Stokes Flow. This example considers the drag minimization of an object subject to a Stokes flow in two dimensions. This problem has a known analytical solution first presented in [31]. The drag is measured by the dissipation of kinetic energy into heat, that is

\[ J_S = \int_{\Omega} \sum_{i,j=1}^{2} \left( \frac{\partial u_i}{\partial x_j} \right)^2 \, dx, \]

where \( \frac{\partial u_i}{\partial x_j} \) denotes the derivative of the i-th velocity component in the j-th direction. The domain consists of a unit square excluding an obstacle as shown in Figure 12 (a). The trivial solution to this problem would be to remove the object from the Stokes-flow completely. This is avoided by introducing additional constraints on the area and centroid of the obstacle. Denoting the target centroid and area of the obstacle as \((C_x, C_y) = (0.5, 0.5)\) and the \(V_O = 0.05\) respectively, we enforce constraints with quadratic penalty terms, yielding the cost functional

\[ J_V = \gamma_1 \left( |\Omega| - |\Omega_0| \right)^2, \]
\[ J_{C_x} = \gamma_2 (C_x - C_{x0})^2, \]
\[ J_{C_y} = \gamma_2 (C_y - C_{y0})^2, \]
\[ J = J_S + J_V + J_{C_x} + J_{C_y}, \]

with penalty parameters \(\gamma_1 > 0\) and \(\gamma_2 > 0\). We denote the target fluid area as \(|\Omega_0| = 1 - V_O = 0.95\), the actual fluid area as \(|\Omega| = \int_{\Omega} 1 \, dx\), and the coordinate of the obstacle’s centroid as \(C_x\) and \(C_y\), for instance \(C_x = \left( \frac{1}{2} - \int_{\Omega} x \, dx \right) / (1 - |\Omega|)\).

To summarize, we can write the optimization problem as

\[ \min_{\Omega, u} J(\Omega, u) \]
subject to

\[-\Delta u + \nabla p = 0 \quad \text{in } \Omega,\]
\[\nabla \cdot u = 0 \quad \text{in } \Omega,\]
\[u = 0 \quad \text{on } \Gamma_2,\]
\[u = u_0 \quad \text{on } \Gamma_1 \cup \Gamma_3,\]
\[\frac{\partial u}{\partial n} + pn = 0 \quad \text{on } \Gamma_4,\]

where $p$ is the fluid pressure, $u_0$ a prescribed boundary velocity, and the domain $\Omega$ is a function of $\Gamma_2$. The boundaries $\Gamma_i, i = 1, \ldots, 4$ are visualized in Figure 12(a). The problem was solved using two overlapping domains as visualized in Figure 12(b). The annulus describing the front mesh, visualized in Figure 12(b) was chosen to have a width of at least three cells of the background mesh. This is important so that one can identity holes added in the geometry by setting cells to be covered (See Figure 5). An interesting effect of this choice is that the front mesh scales with the mesh size, as shown in Figure 13. This means that when we refine the mesh, the number of elements that needs to be deformed does not depend on the total number of degrees of freedom of the MultiMesh. This means that employing the same deformation scheme on the front mesh would be much more efficient than employing it on the whole domain.

We used the variational formulation of the Stokes equations for two overlapping domains, as derived and discretized in [22]. The shape sensitivity of $J_s$ has been derived in [33] and is

\[d J_s(\Omega, u, p)[s] = \int_{\Gamma_2} -s \cdot n \left( \frac{\partial u}{\partial n} \cdot \frac{\partial u}{\partial n} \right) \, dS.\]
The shape sensitivity of $J_V$ is obtained by applying the product rule and Theorem 4.1:

$$dJ_V(\Omega)[s] = -2\gamma_1(\|\Omega\| - |\Omega_0|) \int_{\Gamma_2} s \cdot n \, dS.$$  

(6.10)

Similarly, the shape sensitivity of $J_C$ is obtained using the quotient rule:

$$dJ_C(\Omega)[s] = 2\gamma_2 \frac{1}{|\Omega|} \int_{\Gamma_2} s \cdot n (C_x - x)(C_x - C_{x0}) \, dS.$$  

(6.11)

Similar result can be derived for $dJ_Cy$. Combining (6.9)–(6.11) and obtain the shape sensitivity

$$dJ(\Omega, u, p)[s] = \int_{\Gamma_2} s \cdot n \left( -\left( \frac{\partial u}{\partial n} \cdot \frac{\partial u}{\partial n} \right) - 2\gamma_1(\|\Omega\| - |\Omega_0|) 

+ 2\gamma_2 \frac{1}{|\Omega|} \left( (C_x - x)(C_x - C_{x0}) + (C_y - y)(C_y - C_{y0}) \right) \right) \, dS.$$  

(6.12)

We note that (6.12) does not depend on the adjoint solution. Hence, for this example one does not have to compute the solution of the adjoint equations.

6.3.1. Results. Both a traditional FEM and a MultiMesh FEM Stokes solver was implemented with FEniCS [1, 24]. The meshes used for the experiments are shown in Figure 13. Both solvers use a Taylor-Hood element pair for the discretization, i.e. second order piece-wise continuous polynomials for the velocity and first order piece-wise continuous polynomials for the pressure. The arising linear systems were solved using the direct solver MUMPS [2]. For finer discretizations, it would be beneficial to employ an iterative solver, though efficient preconditioning for the MultiMesh variational problem has not been properly explored. The following results use penalty values of $\beta = 6$ in the variational problem defined in [22], if not otherwise stated.

First, the convergence rates of the Stokes solvers were verified using the manufactured solution given in [22]. For this setup, we expect third and second order convergence rates for velocity and pressure, respectively. The results, listed in Table 4, show that both solvers achieve the expected convergence-rates.

Next, the shape sensitivity was verified using a Taylor test, as shown in Table 5. The expected convergence rate, 2, is obtained. Further tests in random perturbation directions showed similar convergence rates. We also performed Taylor tests on meshes with different resolution, which revealed that the convergence rate reduces on very coarse meshes due to the discrete inconsistency of the shape sensitivity, see Section 4.

Finally, we solved the full shape optimization problem. To ensure that the volume and centroid constraints are sufficiently satisfied, we solved a sequence of optimization problems, with increasing penalty coefficients $\gamma_1$ and $\gamma_2$. The optimized mesh of the previous problem is used as an initial mesh for the next optimization problem. Starting with $\gamma_1 = 10^5$ and $\gamma_2 = 10^3$, five optimization problems were solved in which each $\gamma_1$ and $\gamma_2$ were doubled. The number of cut and uncut cells in the initial MultiMesh was 27490 cells. After a total of 359 optimization iterations, we obtain the final configuration, see Figure 14. The functional dropped from initially 22.5 to 18.4. We observe that the final shape is visually in agreement with [31], which states that the
Fig. 13. a) The traditional FEM mesh and b) the MultiMesh FEM meshes for verification of the Stokes solver. For the MultiMesh approach, the front mesh is scaled to such that the width of the annulus is equivalent to three background cells. In this setup, no extra cells are marked as covered, and there are two interfaces, as shown in Figure 5a).

Table 4
Error and convergence rates for the Stokes problem. The expected convergence-rates are achieved.

|                  | Max Mesh size | $L^2$-error in $u$ | Rate $u$ | $L^2$-error in $p$ | Rate $p$ |
|------------------|---------------|--------------------|----------|--------------------|----------|
| **Traditional FEM** |               |                    |          |                    |          |
| 0.088            | 1.378e-03     | -                  | 7.390e-03| -                  |
| 0.044            | 1.693e-04     | 3.025              | 1.716e-03| 2.107              |
| 0.022            | 2.108e-05     | 3.005              | 4.140e-04| 2.051              |
| 0.011            | 2.635e-06     | 3.000              | 1.018e-04| 2.024              |
| 0.006            | 3.295e-07     | 2.999              | 2.526e-05| 2.011              |
| **MultiMesh FEM** |               |                    |          |                    |          |
| 0.088            | 8.138e-04     | -                  | 1.536e-02| -                  |
| 0.045            | 1.049e-04     | 3.165              | 2.362e-03| 2.884              |
| 0.023            | 1.505e-05     | 3.268              | 4.583e-04| 2.518              |
| 0.012            | 1.560e-06     | 3.076              | 1.061e-04| 2.162              |
| 0.006            | 1.935e-07     | 3.183              | 2.522e-05| 2.192              |

front and back angle of the object should be 90 degrees. We measured the front and back angle of our optimized solution to be 87° and 84° respectively. The mesh deformation was performed using the advection scheme (5.4). In addition, a centroidal Voronoi tessellation [13] (CVT) was used on $\Gamma_2$ to preserve the mesh quality near the the front and back wedge. For the initial mesh, the minimum cell radius ratio was 0.58, where an equilateral triangle has measure 1. For the optimized mesh, this had decreased to 0.32. Compared to a standard FEM, with a mesh with 25420 cells, an iteration of the optimization algorithm is approximately two thirds with the MultiMesh FEM, compared to standard approach. While the assembly of the linear system for the MultiMesh problem is 4 times slower than for the traditional FEM (1.4 seconds to 0.35 seconds), the deformation of the domain is over 20 times faster (0.35 seconds to 7.85 seconds) for the MultiMesh FEM compared to the traditional FEM. In these timings, we have excluded the execution time of the CVT, as they should be
SHAPE OPTIMIZATION USING MULTIMESH FEM WITH NITSCHE COUPLING

Table 5
Results of the Taylor tests for the deformation of an obstacle in a Stokes-flow. The first and second order residuals are defined as $R_0 = |J(\Omega(\epsilon)|s)) - J(\Omega)|$, $R_1 = |J(\Omega(\epsilon)|s)) - J(\Omega) - \epsilon dJ(\Omega)|s))|$. The table show the results for $s = -(1.2e + 4\sin(6\pi x) + 1e4\cos(0.1\pi y))n$.

| $\epsilon$    | $R_0(\epsilon)$ | order | $R_1(\epsilon)$ | order |
|---------------|-----------------|-------|-----------------|-------|
| 5.000e-06     | 3.230e+02       | -     | 3.099e+02       | -     |
| 2.500e-06     | 6.498e+01       | 2.31  | 5.844e+01       | 2.41  |
| 1.250e-06     | 1.579e+01       | 2.04  | 1.251e+01       | 2.22  |
| 6.250e-07     | 4.521e+00       | 1.80  | 2.886e+00       | 2.12  |
| 3.125e-07     | 1.511e+00       | 1.58  | 6.930e-01       | 2.06  |

equal for two boundaries with similar resolution. The mesh quality of the optimized traditional FEM solution was 0.37.

Fig. 14. The flow around the initial obstacle. (b) Flow-profile of the fluid around the optimized obstacle. A front and back-angle of 90 degrees are obtained, as proven analytically in Pironneau [31].

6.4. Orientation of 25 objects in Stokes-flow. As a final example, we considered the problem of optimally rotate 25 obstacles in Stokes flow to minimize dissipation of energy. We consider 25 identical objects placed in a structured fashion, as shown in Figure 15a). There are two identical inlets, with parabolic inlet profiles, and one outlet, that is 66% of total inlet width. The optimization was performed using a MultiMesh consisting of a total of 26 meshes, where each obstacle was represented by a separate mesh, and 22,741 cut and uncut elements. The stopping criteria of the optimization algorithm was set to $\frac{(J(\Omega^{k+1}) - J(\Omega^k))}{J(\Omega^{k+1})} < 10^{-5}$ and achieved after 50 iterations. The optimized configuration is shown in Figure 15b).
7. Concluding remarks. In this paper we have combined known shape optimization techniques, with finite element methods on multiple overlapping meshes. The key features of this approach has been discussed. For shape optimization problem where the change of the domain can be parameterized as a translation or rotation, we observe that the function space of the mesh transformation can be reduced to constant functions. This yields a big speed-up for updating the mesh, where one avoids deformation equations or remeshing techniques. For problems where a part of the domain can be deformed, we have showed that by choosing a meshing describing the part that is changing, one can yield big speed-ups in the mesh deformation step.

Nevertheless, since the MultiMesh FEM is a fairly new method, it has only been explored for time-independent heat and Stokes equation. Further study of Nitsche enforcement of interface conditions is needed to be able to provide stable finite element schemes for overlapping meshes for other equations.

In conclusion, the results reported in this paper, shows that the combination of shape optimization holds great promise as a powerful method for avoiding deformation equations and re-meshing. In a later paper, we will extend this approach to time-dependent problems, with more complex state-equations.

Acknowledgments. The authors would like to acknowledge all FEniCS developers for feedback, and especially those involved in the MultiMesh project.

REFERENCES

[1] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells, The fenics project version 1.5, Archive of Numerical Software, 3 (2015), pp. 9–23.

[2] P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L’Excellent, A fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM Journal on Matrix Analysis and Applications, 23 (2001), pp. 15–41.

[3] L. Armijo, Minimization of functions having lipschitz continuous first partial derivatives, Pacific Journal of mathematics, 16 (1966), pp. 1–3.
SHAPE OPTIMIZATION USING MULTIMESH FEM WITH NITSCHE COUPLING

4. D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, *Unified analysis of discontinuous galerkin methods for elliptic problems*, SIAM journal on numerical analysis, 39 (2002), pp. 1749–1779.

5. K. Bandara, T. Rüberg, and F. Cirak, *Shape optimisation with multiresolution subdivision surfaces and immersed finite elements*, Computer Methods in Applied Mechanics and Engineering, 300 (2016), pp. 510–539.

6. J. Benek, J. Steger, and F. C. Dougherty, *A flexible grid embedding technique with application to the euler equations*, in 6th Computational Fluid Dynamics Conference Danvers, 1983, p. 1944.

7. M. Berggren, *A unified discrete–continuous sensitivity analysis method for shape optimization*, in Applied and numerical partial differential equations, Springer, 2010, pp. 25–39.

8. A. Bernland, E. Wadbro, and M. Berggren, *Acoustic shape optimization using cut finite elements*, International Journal for Numerical Methods in Engineering.

9. P. T. Boggs, A. Althusler, A. R. Laheleire, E. J. Walsh, R. L. Clay, and M. F. Hardwick, *Dart system analysis*, tech. report, Sandia National Laboratories, 2005.

10. E. Burman, S. Claus, P. Hansbo, M. G. Larson, and A. Massing, *CutFEM: Discretizing geometry and partial differential equations*, International Journal for Numerical Methods in Engineering, 104 (2015), pp. 472–501.

11. E. Burman, D. Elfverson, P. Hansbo, M. G. Larson, and K. Larsson, *Shape optimization using the cut finite element method*, Computer Methods in Applied Mechanics and Engineering, 328 (2018), pp. 242–261.

12. M. C. Delfour and J.-P. Zolésio, *Shapes and geometries: metrics, analysis, differential calculus, and optimization*, SIAM, 2011.

13. Q. Du, V. Faber, and M. Gunzburger, *Centroidal voronoi tessellations: Applications and algorithms*, SIAM review, 41 (1999), pp. 637–676.

14. A. Geuzaine and J.-F. Remacle, *Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities*, International journal for numerical methods in engineering, 79 (2009), pp. 1309–1331.

15. C. V. Hansen, A. Logg, and C. Lundholm, *Simulation of flow and view with applications in computational design of settlement layouts*, arXiv preprint arXiv:1610.02277, (2016).

16. H. Harbrecht and F. Loos, *Optimization of current carrying multicables*, Computational Optimization and Applications, 63 (2016), pp. 237–271, https://doi.org/10.1007/s10589-015-9764-2.

17. W. Henshaw, *Overture: An object-oriented framework for overlapping grid applications*, in 32nd AIAA Fluid Dynamics Conference and Exhibit, 2002, p. 3189.

18. M. Hinze, R. Pinnow, M. Ulbrich, and S. Ulbrich, *Optimization with PDE constraints*, vol. 23, Springer Science & Business Media, 2008.

19. A. Johansson, B. Kehlet, M. G. Larson, and A. Logg, *Multimesh finite element methods I: Solving pdes on multiple intersecting meshes*, arXiv preprint arXiv:1805.08741, (2018).

20. A. Johansson, B. Kehlet, and A. Logg, *Construction of quadrature rules on general polygonal and polyhedral domains in cut finite element methods*, In preparation, (2018).

21. A. Johansson and M. G. Larson, *A high order discontinuous Galerkin Nitsche method for elliptic problems with fictitious boundary*, Numerische Mathematik, 123 (2013), pp. 607–628.

22. A. Johansson, M. G. Larson, and A. Logg, *High order cut finite element methods for the Stokes problem*, Advanced Modeling and Simulation in Engineering Sciences, 2 (2015), p. 1.

23. W. Lim and H. M. Tsai, *Aerodynamic design optimization by the adjoint equation method on overset grids*, in 44th AIAA Aerospace Sciences Meeting and Exhibit, 2006, p. 54.

24. A. Logg and G. N. Wells, *Dolfin: Automated finite element computing*, ACM Transactions on Mathematical Software (TOMS), 37 (2010), p. 20.

25. F. Loos, *Joule Heating in Connecting Structures of Automotive Electric Devices—Modelling, Simulation and Optimization*, PhD thesis, München, Univ. der Bundeswehr, Diss., 2014, 2014.

26. B. Mohammadi and O. Pironneau, *Applied shape optimization for fluids*, Oxford university press, 2010.

27. A. R. Najafi, M. Safdari, D. A. Tortorelli, and P. H. Geubelle, *A gradient-based shape optimization scheme using an interface-enriched generalized fem*, Computer Methods in Applied Mechanics and Engineering, 296 (2015), pp. 1 – 17, https://doi.org/https://doi.org/10.1016/j.cma.2015.07.024.

28. E. J. Nielsen and M. A. Park, *Using an adjoint approach to eliminate mesh sensitivities in computational design*, AIAA journal, 44 (2006), p. 948.

29. J. Nitsche, *Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung
von Teilräumen, die keinen Randbedingungen unterworfen sind, in Abhandlungen aus dem mathematischen Seminar der Universität Hamburg, vol. 36, Springer, 1971, pp. 9–15.

[30] L. Noël and P. Duysinx, Shape optimization of microstructural designs subject to local stress constraints within an xfem-level set framework, Structural and Multidisciplinary Optimization, 55 (2017), pp. 2323–2338.

[31] O. Pironneau, On optimum design in fluid mechanics, Journal of Fluid Mechanics, 64 (1974), pp. 97–110.

[32] J. J. Reuther, A. Jameson, J. J. Alonso, M. J. Rimlinger, and D. Saunders, Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, part I, Journal of Aircraft, 36 (1999), pp. 51–60.

[33] S. Schmidt, Efficient large scale aerodynamic design based on shape calculus, 2010.

[34] S. Schmidt, A Two Stage CVT/Eikonal Convection Mesh Deformation Approach for Large Nodal Deformations, arXiv preprint arXiv:1411.7663, (2014).

[35] S. Schmidt, C. Ilic, V. Schulz, and N. R. Gauger, Three-dimensional large-scale aerodynamic shape optimization based on shape calculus, AIAA journal, (2013).

[36] S. Schmidt, E. Wadbro, and M. Berggren, Large-scale three-dimensional acoustic horn optimization, SIAM Journal on Scientific Computing, 38 (2016), pp. B917–B940.

[37] J. Sokolowski and J.-P. Zolesio, Introduction to shape optimization, in Introduction to Shape Optimization, Springer, 1992, pp. 5–12.

[38] R. Stangl, Ein Euler-Verfahren zur Berechnung der Strömung um einen Hubschrauber im Vorwärtsflug, Herbert Utz Verlag, 1996.

[39] G. Starius, Composite mesh difference methods for elliptic boundary value problems, Numerische Mathematik, 28 (1977), pp. 243–258.

[40] K. Stein, T. Tezduyar, and R. Benney, Mesh moving techniques for fluid-structure interactions with large displacements, Journal of applied mechanics, 70 (2003), pp. 58–63.

[41] E. Tekin and I. Sabuncuoglu, Simulation optimization: A comprehensive review on theory and applications, IIE transactions, 36 (2004), pp. 1067–1081.

[42] L. Van Miegroet, N. Moës, C. Fleury, and P. Duysinx, Generalized shape optimization based on the level set method, in Proceedings of the 6th World Congress of Structural and Multidisciplinary Optimization (WCSMO6), 2005.

[43] C. H. Villanueva and K. Maute, Cutfem topology optimization of 3d laminar incompressible flow problems, Computer Methods in Applied Mechanics and Engineering, 320 (2017), pp. 444–473.

[44] E. Volkov, The method of composite meshes, Automatic Programming, Numerical Methods and Functional Analysis, 96 (1970), p. 145.