

HIGHER-ORDER MOVING MESH METHODS FOR PDE-CONSTRAINED SHAPE OPTIMIZATION*

ALBERTO PAGANINI[†], FLORIAN WECHSUNG[†], AND PATRICK E. FARRELL[†]

Abstract. We present a new approach to discretizing shape optimization problems that generalizes standard moving mesh methods to higher-order mesh deformations and that is naturally compatible with higher-order finite element discretizations of PDE-constraints. This shape optimization method is based on discretized deformation diffeomorphisms and allows for arbitrarily high resolution of shapes with arbitrary smoothness. Numerical experiments show that this method allows the solution of PDE-constrained shape optimization problems to high accuracy.

Key words. shape optimization, PDE-constraint, finite elements, moving mesh

AMS subject classifications. 49Q10, 65N30

DOI. 10.1137/17M1133956

1. Introduction. Shape optimization problems are optimization problems where the control to be optimized is the shape of a domain. Their basic formulation generally reads

$$(1) \quad \text{find } \Omega^* \in \underset{\Omega \in \mathcal{U}_{\text{ad}}}{\operatorname{argmin}} \mathcal{J}(\Omega),$$

where \mathcal{U}_{ad} denotes a collection of *admissible shapes* and $\mathcal{J} : \mathcal{U}_{\text{ad}} \rightarrow \mathbb{R}$ represents a *shape functional*. In many applications, the shape functional depends not only on the shape of a domain $\Omega \subset \mathbb{R}^d$ but also on the solution u of a boundary value problem (BVP) posed on Ω , in which case (1) becomes

$$(2a) \quad \text{find } \Omega^* \in \underset{\Omega \in \mathcal{U}_{\text{ad}}}{\operatorname{argmin}} \mathcal{J}(\Omega, u_\Omega) \quad \text{subject to}$$

$$(2b) \quad u_\Omega \in V(\Omega), \quad a_\Omega(u_\Omega, v) = f_\Omega(v) \quad \text{for all } v \in W(\Omega),$$

where (2b) represents the variational formulation of a BVP that acts as a PDE-constraint.

These problems are said to be *PDE-constrained* and are notoriously difficult to solve because the dependence of \mathcal{J} on the domain is nonconvex. Additionally, the function u_Ω cannot be computed analytically. Even approximating it with a numerical method is challenging because the computational domain of the PDE-constraint is the unknown variable to be solved for in the shape optimization problem.

The literature abounds with numerical methods for BVPs. Here, we consider approximation by means of finite elements, which has become the most popular choice for PDE-constrained shape optimization due to its flexibility for engineering applications.

*Submitted to the journal's Methods and Algorithms for Scientific Computing section June 9, 2017; accepted for publication (in revised form) May 7, 2018; published electronically August 2, 2018.

<http://www.siam.org/journals/sisc/40-4/M113395.html>

Funding: This work was supported by EPSRC grants EP/K030930/1 and EP/M011151/1 and by the EPSRC Centre for Doctoral Training in Industrially Focused Mathematical Modelling (EP/L015803/1) in collaboration with London Computational Solutions.

[†]Mathematical Institute, University of Oxford, Oxford, OX2 6GG, UK (paganini@maths.ox.ac.uk, wechsung@maths.ox.ac.uk, patrick.farrell@maths.ox.ac.uk).

Nevertheless, it is worth mentioning that alternatives based on other discretizations have also been considered [6, 9, 20, 53].

Most commonly, PDE-constrained shape optimization problems are formulated with the aim of further improving the performance of an initial design Ω^0 . The standard procedure to pursue this goal is to iteratively update some parametrization of Ω^0 to decrease the value of \mathcal{J} . Obviously, the choice of this parametrization has an enormous influence on the design of the related shape optimization algorithm and on the search space \mathcal{U}_{ad} itself.

In this work, we parametrize shapes by applying deformation diffeomorphisms to the initial guess Ω^0 . In this framework, solving a shape optimization problem translates into constructing an optimal diffeomorphism. To construct this diffeomorphism with numerical methods, we introduce a discretization of deformation vector fields.

Performing shape optimization by optimizing deformations is not new [23, 24, 25, 35, 43]. The novelty of our work is to fuse this strategy with finite element discretizations that use a reference element in such a way that standard finite element software can be reused without modification to discretize the state constraints. Our strategy can employ distinct discretizations for the control and state where it is advantageous to do so. In particular, deformations can be discretized with B-splines, Lagrangian finite elements, or harmonic functions, among others. This discretization can comprise arbitrarily many basis functions and thus allows for arbitrarily high resolution of shapes with arbitrary smoothness. The result is a higher-order generalization of moving mesh methods [1, 14] that is straightforward to implement.

There are several advantages to using higher-degree and smoother transformations. First, higher-degree parametrization of domains allows for the consideration of more general shapes (beyond polytopes). Secondly, the efficiency of a higher-order discretization of a BVP hinges on the regularity of its solution, which depends on the regularity of the computational domain, among other factors. Finally, a smoother discretization of deformation vector fields allows the computation of more accurate Riesz representatives of shape derivatives [36, 48] and thus, more accurate descent directions for shape optimization algorithms.

The rest of this article is organized as follows. In section 2, we describe how we model the search space \mathcal{U}_{ad} with deformation diffeomorphisms and discuss the advantages and disadvantages of this choice. In section 3, we give a brief introduction to *shape calculus* and explain how to compute steepest descent updates for deformation diffeomorphisms using shape derivatives of shape functionals. In section 4, we emphasize that having a PDE-constraint necessitates the solution of a BVP and its adjoint at each step of the optimization, and we comment on the error introduced by their finite element discretizations. In section 5, we give an introduction to isoparametric finite elements and explain why it is natural to employ this kind of discretization to approximate the state variable u_Ω when the domain Ω is modified by a diffeomorphism. In section 6, we examine implementation aspects of the algorithm resulting from section 3 and section 5. In particular, we give detailed remarks for an efficient implementation of a decoupled discretization of the state and control variables. Finally, in section 7, we perform numerical experiments. First, we consider a well-posed test case and investigate the impact of the discretization of the state and of the control variables on the performance of higher-order moving mesh methods, showing that these methods can be employed to solve PDE-constrained shape optimization problems to high accuracy. Second, we consider a more challenging PDE-constrained shape optimization problem and show that increasing the degree of transformations can be beneficial.

Remark 1. Shape optimization problems with a distinction between computational domain and control variable also exist. For instance, this is the case for PDE-constrained optimal control problems where the control is a piecewise constant coefficient in the PDE-constraint [41, 47], in which case the control is the shape of the contour levels of the piecewise constant coefficient. The approach suggested in this work extends to this more general type of shape optimization problem. However, to simplify the discussion and reduce the amount of technicalities, we consider only problems of the form (2).

2. Parametrization of shapes via diffeomorphisms. Among the many possibilities for defining \mathcal{U}_{ad} , we choose to construct it by collecting all domains that can be obtained by applying (sufficiently regular) geometric transformations¹ to an initial domain Ω^0 , that is,

$$(3) \quad \mathcal{U}_{\text{ad}} := \{T(\Omega^0) : T \in \mathcal{T}_{\text{ad}}\},$$

where \mathcal{T}_{ad} is (a subgroup of) the group of $W^{1,\infty}$ -diffeomorphisms. We recall that $W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$ is the Sobolev space of locally integrable vector fields with essentially bounded weak derivatives. We impose this regularity requirement on \mathcal{T}_{ad} to guarantee that the state constraint (2b) is well-defined for every domain in \mathcal{U}_{ad} (assuming that it is well-defined on the initial domain Ω^0). Note that it may be necessary to strengthen the regularity requirements on \mathcal{T}_{ad} if the PDE-constraint is a BVP of higher order, such as, for example, the biharmonic equation [10].

While there are many alternatives (for instance, level sets [2] or phase fields [27]), we prefer to describe \mathcal{U}_{ad} as in (3) because it incorporates an explicit description of the boundaries of the domains contained within it. In fact, describing shapes via diffeomorphisms is a standard approach in shape optimization; cf. [17, Chap. 3]. From a theoretical point of view, it is possible to impose a metric on (3) and to investigate the existence of optimal solutions within this framework [44]. Recently, the convergence of Newton's method in this framework has also been investigated [57].

Remark 2. The representation of domains via transformations in (3) is not unique, and it is generally possible to find two transformations $T_1 \in \mathcal{T}_{\text{ad}}$ and $T_2 \in \mathcal{T}_{\text{ad}}$ such that $T_1 \neq T_2$ and $T_1(\Omega^0) = T_2(\Omega^0)$. For instance, this is the case if Ω^0 is a ball and $T_2 = T_1 \circ T_R$, where T_R is a rotation around the center of Ω^0 . To obtain a one-to-one correspondence between shapes and transformations, one can introduce equivalence classes, but this is not particularly relevant for this work.

To shorten the notation, we introduce the reduced functional

$$(4) \quad j : \mathcal{U}_{\text{ad}} \rightarrow \mathbb{R}, \quad \Omega \mapsto \mathcal{J}(\Omega, u_\Omega),$$

which is well-defined under the following assumption on the PDE-constraint (2b).

Assumption 1. Henceforth, we assume that the BVP (2b) that acts as PDE-constraint is well-defined in the sense of Hadamard: for every $\Omega \in \mathcal{U}_{\text{ad}}$ the BVP (2b) has a unique solution u_Ω that depends continuously on the BVP data.

In section 1, we mentioned that shape optimization problems are solved updating iteratively some parametrization of Ω^0 , that is, constructing a sequence of domains

¹A geometric transformation is a bijection from \mathbb{R}^d onto itself.

$\{\Omega^{(k)}\}_{k \in \mathbb{N}}$ so that $\{j(\Omega^{(k)})\}_{k \in \mathbb{N}}$ decreases monotonically. For simplicity, we relax the terminology and call such a sequence *minimizing*, although the equality

$$(5) \quad \lim_{k \rightarrow \infty} j(\Omega^{(k)}) = \inf_{\Omega \in \mathcal{U}_{\text{ad}}} j(\Omega)$$

may not be satisfied.

When the search space \mathcal{U}_{ad} is constructed as in (3), computing $\{\Omega^{(k)}\}_{k \in \mathbb{N}}$ translates into creating a sequence of diffeomorphisms $\{T^{(k)}\}_{k \in \mathbb{N}}$. Generally, the sequence $\{T^{(k)}\}_{k \in \mathbb{N}}$ is constructed according to the following procedure:

- (a) given the current iterate $T^{(k)}$, derive a tentative iterate \tilde{T} ;
- (b) if \tilde{T} satisfies certain quality criteria, set $T^{(k+1)} = \tilde{T}$ and move to the next step; otherwise compute another \tilde{T} .

In the next section, we discuss the computation of \tilde{T} with shape derivatives. For simplicity, we first assume that the state variable u_Ω is known analytically and restrict our considerations to the reduced functional j defined in (4). The role of the PDE-constraint and the discretization of the state variable is discussed in section 4.

3. Iterative construction of diffeomorphisms. Shape calculus offers an elegant approach for constructing a minimizing sequence of domains $\{\Omega^{(k)}\}_{k \in \mathbb{N}}$. The key tool is the derivative of the shape functional \mathcal{J} with respect to shape perturbations. To give a more precise description, let us first introduce the operator

$$(6) \quad J : \mathcal{T}_{\text{ad}} \rightarrow \mathbb{R}, \quad T \mapsto j(T(\Omega^0)).$$

Since $\mathcal{T}_{\text{ad}} \subset W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$, which is a Banach space with respect to the norm [21, sect. 5.2.2]

$$(7) \quad \|T\|_{W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)} := \sum_{|\alpha| \leq 1} \text{ess sup} \|\mathbf{D}^\alpha T\|,$$

we can formally define the directional derivative of J at $T \in \mathcal{T}_{\text{ad}}$ in the direction $\mathcal{T} \in W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$ through the limit

$$(8) \quad dJ(T; \mathcal{T}) := \lim_{s \rightarrow 0^+} \frac{J((\mathcal{I} + s\mathcal{T}) \circ T) - J(T)}{s} = \lim_{s \rightarrow 0^+} \frac{J(T + s\mathcal{T} \circ T) - J(T)}{s}.$$

Remark 3. Note that $\mathcal{I} + s\mathcal{T}$ is a $W^{1,\infty}$ -diffeomorphism for sufficiently small s [1, Lem. 6.13].

A shape functional \mathcal{J} is said to be *shape differentiable* (in $T(\Omega^0)$) if the corresponding functional (6) is Fréchet differentiable (in T), that is, if (8) defines a linear continuous operator on $W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$ such that

$$(9) \quad |J(T + s\mathcal{T} \circ T) - J(T) - dJ(T; s\mathcal{T})| = o(s) \quad \text{for all } \mathcal{T} \in W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d).$$

Remark 4. Generally, Assumption 1 is not sufficient to guarantee that \mathcal{J} is shape differentiable. In particular, it is necessary to ensure that the solution operator $\Omega \mapsto u_\Omega$ is continuously differentiable; cf. [34, sect. 1.6].

The Fréchet derivative dJ can be used to construct a sequence of diffeomorphisms $\{T^{(k)}\}_{k \in \mathbb{N}}$ to solve (2) in a steepest descent fashion. More specifically, the entries of this sequence take the form

$$(10) \quad T^{(0)}(\mathbf{x}) = \mathbf{x} \quad \text{and} \quad T^{(k+1)}(\mathbf{x}) = (\mathcal{I} + dT^{(k)}) \circ (T^{(k)}(\mathbf{x})),$$

where the update $dT^{(k)} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is computed with the help of dJ . For instance, we could define [34, p. 103]

$$(11) \quad dT^{(k)} \in \underset{\substack{\mathcal{T} \in W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d), \\ \|\mathcal{T}\|_{W^{1,\infty}} = 1}}{\operatorname{argmin}} dJ(T^{(k)}; \mathcal{T}).$$

Unfortunately, such a descent direction may not exist without making further assumptions on dJ because $W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$ is not reflexive; cf. [39]. However, in [41] it has been shown that in many instances (and under suitable assumptions), the operator dJ takes the form

$$(12) \quad dJ(T^{(k)}; \mathcal{T}) = \int_{T^{(k)}(\Omega_0)} \sum_{i,j=1}^d s_1^{i,j} \mathbf{D}\mathcal{T}^{i,j} + \sum_{\ell=1}^d s_0^\ell \mathcal{T}^\ell \, d\mathbf{x},$$

where $s_1^{i,j}$, $i, j = 1, \dots, d$, and s_0^ℓ , $\ell = 1, \dots, d$, are (instance dependent) $L^1(\mathbb{R}^d)$ -functions. The following proposition² states that, in this case, (11) can be used to define steepest descent directions.

PROPOSITION 3.1. *Let dJ be as in (12). Then, there exists a descent direction $dT^{(k)}$ as defined in (11).*

Proof. First of all, we recall that $L^\infty(D)$ is isometrically isomorphic to the dual X^* of $X = L^1(D)$ (for any open domain $D \subset \mathbb{R}^m$ in any fixed dimension m). We denote by $\phi_D : L^\infty(D) \rightarrow X^*$ this isomorphism. The duality pairing $\langle \cdot, \cdot \rangle_{X^* \times X}$ can be characterized by

$$(13) \quad \langle f, g \rangle_{X^* \times X} = \int_D \phi_D^{-1}(f)g \, d\mathbf{x}.$$

Clearly, similar pairings exist for Cartesian products of $L^\infty(D)$. Finally, note that $L^1(D)$ is separable. Therefore, by the Banach–Alaoglu theorem, any bounded sequence in $L^\infty(D)$ has a subsequence $\{\mathbf{x}_n\}$ that converges weakly- $*$ to an $\mathbf{x} \in L^\infty(D)$, that is,

$$(14) \quad \lim_{n \rightarrow \infty} \int_D \mathbf{x}_n g \, d\mathbf{x} = \int_D \mathbf{x} g \, d\mathbf{x} \quad \text{for every } g \in L^1(D).$$

Using these results, we show that a steepest descent direction exists.

Let \mathcal{T}_n be a minimizing sequence of (11). By definition, \mathcal{T}_n is bounded in $W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$, and hence in $L^\infty(\mathbb{R}^d; \mathbb{R}^d)$, too. Thus, there exists a subsequence \mathcal{T}_{n_k} that converges weakly- $*$ to a $T \in L^\infty(\mathbb{R}^d; \mathbb{R}^d)$. Since \mathcal{T}_{n_k} is bounded in $W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$, there is a subsequence $\mathcal{T}_{n_{k_\ell}}$ such that $\mathbf{D}\mathcal{T}_{n_{k_\ell}}$ converges weakly- $*$ in $L^\infty(\mathbb{R}^d; \mathbb{R}^{d,d})$. By the definition of weak derivative, it is easy to see that the weak- $*$ limit of $\mathbf{D}\mathcal{T}_{n_{k_\ell}}$ is $\mathbf{D}T$. This shows that $T \in W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$.

Since (12) is a sum of duality pairings as in (13) (with $D = \mathbb{R}^d$ and $g = \chi_{T^{(k)}(\Omega_0)} s_1^{i,j}$ or $g = \chi_{T^{(k)}(\Omega_0)} s_0^\ell$, where $\chi_{T^{(k)}(\Omega_0)}$ is the characteristic function associated to $T^{(k)}(\Omega_0)$), dJ is weakly- $*$ continuous. Therefore, T is a minimizer, because it is the weak- $*$ limit of $\mathcal{T}_{n_{k_\ell}}$, which is a subsequence of a minimizing sequence.

²We provide a full proof of this proposition because, to the best of our knowledge, this result is new.

Finally, to show that $\|T\|_{W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)} = 1$, we recall that the norm of a Banach space is weak-* lower semicontinuous. Therefore,

$$(15) \quad \|T\|_{W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)} \leq \liminf_{k \rightarrow \infty} \|\mathcal{T}_{n_k}\|_{W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)} \leq 1,$$

and since $dJ(T^{(k)}; \cdot)$ is linear, $\|T\|_{W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)} = 1$.

To conclude, note that $dJ(T^{(k)}; T) > -\infty$ because $dJ(T^{(k)}; \cdot)$ is continuous. \square

Although possibly well-defined, it is challenging to compute such a descent direction $dT^{(k)}$ (because $W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)$ is infinite dimensional and neither reflexive nor separable). One possible remedy is to introduce a Hilbert subspace $(\mathcal{X}, (\cdot, \cdot)_{\mathcal{X}})$, $\mathcal{X} \subset W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)$ and to compute

$$(16) \quad dT_{\mathcal{X}}^{(k)} := \operatorname{argmin}_{\substack{\mathcal{T} \in \mathcal{X}, \\ \|\mathcal{T}\|_{\mathcal{X}}=1}} dJ\left(T^{(k)}; \mathcal{T}\right).$$

The solution of (16) is parallel to the gradient of dJ with respect to $(\cdot, \cdot)_{\mathcal{X}}$ and, up to a scaling factor, it can be computed by solving the variational problem: find $dT^{(k)}$ such that

$$(17) \quad \left(dT^{(k)}, \mathcal{T}\right)_{\mathcal{X}} = -dJ\left(T^{(k)}; \mathcal{T}\right) \quad \text{for all } \mathcal{T} \in \mathcal{X},$$

which is well-posed by the Riesz representation theorem. However, the condition $\mathcal{X} \subset W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)$ is restrictive (for a Hilbert space). For instance, the general Sobolev inequalities guarantee that the Sobolev space $H^k(\mathbb{R}^d;\mathbb{R}^d)$ is contained in $W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)$ only for $k \geq d/2 + 1$ [21, sect. 5.6.3].

A more popular approach is to introduce a finite dimensional subspace $Q_N \subset \mathcal{X} \cap W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)$ and to compute the solution of

$$(18) \quad \left(dT_N^{(k)}, \mathcal{T}_N\right)_{\mathcal{X}} = -dJ\left(T^{(k)}; \mathcal{T}_N\right) \quad \text{for all } \mathcal{T}_N \in Q_N.$$

In this case, the requirement $\mathcal{X} \subset W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)$ can be dropped as long as the dimension $N := \dim(Q_N)$ of Q_N is finite. However, note that if $\{Q_N\}_{N \in \mathbb{N}}$ is a family of nested finite dimensional spaces such that their union $\cup_{N \in \mathbb{N}} Q_N$ is dense in \mathcal{X} , the sequence $dT_N^{(k)}$ can be interpreted as the Ritz–Galerkin approximation of $dT^{(k)}$. Therefore, as $N \rightarrow \infty$, the sequence $dT_N^{(k)}$ may converge to an element of $\mathcal{X} \setminus W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)$, which does not qualify as an admissible update.

The trial space Q_N can be constructed with linear Lagrangian finite elements defined on (a mesh of) a hold-all domain $D \supset \Omega^0$. The resulting algorithm is equivalent to standard moving mesh methods [48]. Alternatively, one can employ tensorized B-splines [35]. Lagrangian finite elements have the advantage of inclusion in standard finite element software, whereas B-splines offer higher regularity, which is often desirable (as we will argue in section 6). For instance, univariate B-splines of degree d are in $W^{d,\infty}(\mathbb{R})$ [37], whereas Lagrangian finite elements are not even C^1 . As for the Hilbert space \mathcal{X} , one usually opts for $\mathcal{X} = H^1(D)$ or, equivalently, for $H^{1/2}(\partial\Omega^{(k)})$ combined with an elliptic extension operator onto D [55]. This choice can be motivated by considerations of the shape Hessian [20, 55].

To the best of our knowledge, it has not been settled yet which definition of steepest direction among (11), (16), and (18) is best suited to formulate a numerical shape optimization algorithm. Since the focus of this work is more on the discretization of

shape optimization problems than on actual optimization algorithms, we postpone investigations of this topic to future research. In our numerical experiments in section 7, we will employ (18), which is the computationally most tractable definition. However, note that computing steepest directions according to (11) or (16) would also inevitably involve some discretization, because $W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$ (and generally \mathcal{X}) are infinite dimensional.

We conclude this section with the Hadamard homeomorphism theorem [40, Thm. 1.2], which gives explicit criteria to verify that the entries of the sequence $\{T^{(k)}\}_{k \in \mathbb{N}}$ defined in (10) are admissible transformations.

THEOREM 3.2. *Let X and Y be finite dimensional Euclidean spaces, and let $T : X \rightarrow Y$ be a C^1 -mapping that satisfies the following conditions:*

1. $\det(\mathbf{D}T)(x) \neq 0$ for all $x \in X$.
2. $\|T(x)\| \rightarrow \infty$ as $\|x\| \rightarrow \infty$.

Then T is a C^1 -diffeomorphism from X to Y .

A counterpart of Theorem 3.2 for $W^{1,\infty}$ -transformations can be found in [26].

For the sequence (10), note that the second hypothesis of Theorem 3.2 is automatically satisfied if the hold-all domain D is bounded, because the update $dT^{(k)}$ has compact support, and $T^{(k+1)}(\mathbf{x}) = \mathbf{x}$ for every $\mathbf{x} \in \mathbb{R}^d \setminus \overline{D}$.

4. Shape derivatives of PDE-constrained functionals. To simplify the exposition, in the previous section we treated the dependence of J on u implicitly; this dependence was hidden in the reduced functional j . We now examine the consequences of this dependence, as it introduces additional difficulties. Indeed, it is generally the case that to evaluate the Fréchet derivative dJ it is necessary to solve (at least) one BVP. To illustrate this fact, we consider the following example:

$$(19a) \quad \mathcal{J}(\Omega, u_\Omega) = \frac{1}{2} \int_{\Omega} u_\Omega^2 \, d\mathbf{x} \quad \text{subject to}$$

$$(19b) \quad u_\Omega \in H^1(\Omega), \quad \int_{\Omega} \nabla u_\Omega \cdot \nabla v + u_\Omega v \, d\mathbf{x} = \int_{\Omega} v \, d\mathbf{x} \quad \text{for all } v \in H^1(\Omega).$$

Its shape derivative can be computed using function space parametrization [17, Chap. 10, sect. 2.2] and reads [47, eq. 2.12]

$$(20) \quad dJ(T; \mathcal{T}) = \int_{T(\Omega^0)} (\nabla u_{T(\Omega^0)} \cdot (\mathbf{D}\mathcal{T} + \mathbf{D}\mathcal{T}^\top) \nabla p \\ + (p + u_{T(\Omega^0)}^2 - \nabla u_{T(\Omega^0)} \cdot \nabla p - u_{T(\Omega^0)} p) \operatorname{div} \mathcal{T}) \, d\mathbf{x},$$

where $p \in H^1(T(\Omega^0))$ is the solution of the adjoint BVP

$$(21) \quad \int_{T(\Omega^0)} \nabla p \cdot \nabla v + p v \, d\mathbf{x} = \int_{T(\Omega^0)} u_{T(\Omega^0)} v \, d\mathbf{x} \quad \text{for all } v \in H^1(T(\Omega^0)).$$

Formula (20) clearly shows that it is necessary to compute the functions $u_{T(\Omega^0)}$ and p to evaluate dJ . The adjoint BVP (21) is introduced to derive a formula of dJ that does not contain the shape derivative of u_Ω . This is a well-known strategy in PDE-constrained optimization [34, Sect. 1.6].

In general, deriving explicit formulae for Fréchet derivatives of PDE-constrained functionals is a delicate and error prone task. However, in many instances one can introduce a Lagrangian functional that allows the automation of the differentiation process and gives the correct adjoint equations [34, sect. 1.6.4]. The level of automation

is such that numerical software is capable of differentiating several PDE-constrained functionals [22]. Clearly, Lagrangians are useful also for the special case of PDE-constrained shape functionals [17, Chap. 10], and dedicated numerical software for shape differentiation has recently become available [52]. In particular, the latter implements shape calculus formulae from [17, Chap. 10] to compute shape derivatives and uses standard Lagrangian techniques to derive adjoint equations [34].

Remark 5. The Hadamard–Zolésio structure theorem [17, Chap. 9, Thm. 3.6] states that, under certain regularity assumptions on Ω , the Fréchet derivative $dJ(\Omega; \mathcal{T})$ depends only on perturbations $\mathcal{T}(\partial\Omega)$ of the domain boundary. As a consequence, the derivative of most shape functionals can be formulated as an integral both in the volume Ω and on the boundary $\partial\Omega$, and these formulations are equivalent. For instance, the boundary formulation that corresponds to (20) reads [47, eq. 2.13]

$$(22) \quad \int_{T(\partial\Omega^0)} \mathcal{T} \cdot \mathbf{n} \left(u_{T(\Omega^0)}^2 - \nabla u_{T(\Omega^0)} \cdot \nabla p - u_{T(\Omega^0)} p + p \right) dS.$$

When the state and the adjoint variables are replaced by numerical approximations, these two formulae define two different approximations of dJ . In the framework of finite elements, it has been shown that volume based formulations usually offer higher accuracy compared to their boundary based counterparts [36, 46]. Additionally, the combination of volume based formulae with discretization of the control variable results in shape optimization algorithms for which the paradigms *optimize-then-discretize* and *discretize-then-optimize* commute [12, 23, 25, 35]. This does not hold in general for boundary based formulae because piecewise linear finite elements do not fulfill the necessary regularity requirements, and the equivalence of boundary and volume based formulae is not guaranteed [12].

5. Isoparametric Lagrangian finite elements. To evaluate the shape functional $\mathcal{J}(\Omega, u_\Omega)$, it is necessary to approximate the function u_Ω , which is the solution of the PDE-constraint (2b). For the Fréchet derivative dJ , it may be necessary to also approximate the solution p of an adjoint BVP. In this work, we consider the discretization of (2b) and the adjoint BVP by means of finite elements. Finite element spaces are defined on meshes of the computational domain. As shape optimization algorithms modify the computational domain, a new mesh is required at each iteration. This new mesh can be constructed either de novo or by modifying a previously existing mesh. On the one hand, remeshing should be avoided because it is computationally expensive and may introduce undesirable noise in the optimization algorithm (in particular, the value of the shape functional can change after remeshing). On the other hand, updating the mesh is a delicate process and may return a mesh with poor quality (which in turn introduces noise in the optimization as well).

Isoparametric finite elements offer an interesting perspective on the process of mesh updating that fits well with our encoding of changes in the domain via geometric transformations. In particular, with isoparametric finite elements it is possible to mimic the modification of the computational domain without tampering directly with the finite element mesh. Additionally, isoparametric finite element theory provides insight into the extent to which remeshing can be avoided. Next, we provide a concise recapitulation of isoparametric finite element theory. For simplicity, we assume that the PDE-constraint is a linear V -elliptic second-order BVP. However, we believe that most of the considerations readily cover more general BVPs. For a more thorough introduction to isoparametric finite elements, we refer to [15, sect. 4.3].

The Ritz–Galerkin discretization of (2b) reads

$$(23) \quad \text{find } u_h \in V_h(\Omega), \quad a_\Omega(u_h, v_h) = f_\Omega(v_h) \quad \text{for all } v_h \in V_h(\Omega),$$

where $V_h(\Omega)$ is a finite dimensional subspace of the trial space $V(\Omega)$ and of the test space $W(\Omega)$. Henceforth, we restrict ourselves to Lagrangian finite element approximations on simplicial meshes.

Let us assume for the moment that Ω is a polytope. The most common construction of finite element spaces begins with a triangulation $\Delta_h(\Omega)$ of Ω . This triangulation is used to introduce global basis functions that span the finite element space. The finite element space is called *Lagrangian* if the degrees of freedom of its global basis functions are point evaluations [15, p. 36], and it is called *of degree p* if the local basis functions, that is, the restriction of global basis functions to elements K of the triangulation, are polynomials of degree p .

It is well-known that Lagrangian finite elements on simplicial meshes are *affine equivalent*. Affine equivalence means that we can define a reference element \hat{K} and a set of reference local basis functions $\{\hat{b}_i\}_{i \leq M}$ on \hat{K} , and construct a family of affine diffeomorphisms $\{G_K : \hat{K} \rightarrow K\}_{K \in \Delta_h(\Omega)}$ such that the local basis functions $\{b_i^K\}_{i \leq M}$ on K satisfy $b_i^K(\mathbf{x}) = \hat{b}_i(G_K^{-1}(\mathbf{x}))$. Note that both $\{b_i^K\}_{i \leq M}$ and $\{\hat{b}_i\}_{i \leq M}$ are polynomials, because the pullback induced by a bijective affine transformation is an automorphism.

Issues with this construction arise if Ω has curved boundaries. In this case, we introduce first an affine equivalent finite element space $V_h(\hat{\Omega})$ built on the triangulation $\Delta_h(\hat{\Omega})$ of a polytope $\hat{\Omega}$ that approximates Ω . Then, we construct a vector field $F \in (V_h(\hat{\Omega}))^d$ such that $F(\partial\hat{\Omega}) \approx \partial\Omega$ and generate a curved³ triangulation $\Delta_h(\Omega)$ by deforming the elements of $\Delta_h(\hat{\Omega})$ according to F . Finally, we define the finite element space $V_h(\Omega)$ on $\Delta_h(\Omega)$ by choosing $b_i^K(\mathbf{x}) = \hat{b}_i(G_K^{-1}(F^{-1}(\mathbf{x})))$ as local basis functions. This construction leads to so-called isoparametric finite elements. Again, this space is called *Lagrangian* if the reference local basis functions $\{\hat{b}_i\}_{i \leq M}$ are polynomials. However, note that the local basis functions $\{b_i\}_{i \leq M}$ of isoparametric Lagrangian finite elements may not be polynomials.

Isoparametric Lagrangian finite elements on curved domains are proved to retain the approximation properties of Lagrangian finite elements on polytopes under the following additional assumptions [15, Thm. 4.3.4]:

1. the triangulation $\Delta_h(\hat{\Omega})$ is regular [15, p. 124],
2. the mesh width h is sufficiently small,
3. for every quadrature point $\mathbf{x}_q \in \hat{K}$, and for every element $K \in \Delta_h(\hat{\Omega})$,

$$(24) \quad \|F(G_K(\mathbf{x}_q)) - G_K(\mathbf{x}_q)\| = \mathcal{O}(h^p),$$

and $F(G_K(\mathbf{x}_q)) \in \partial\Omega$ whenever $G_K(\mathbf{x}_q) \in \partial\hat{\Omega}$.

Equation (24) is sufficient to guarantee that the map $F \circ G_K$ is a diffeomorphism and to provide algebraic estimates of the form

$$(25) \quad \|\mathbf{D}^\alpha(F \circ G_K)\| = \mathcal{O}(h^\alpha) \quad \text{for } 0 \leq \alpha \leq p,$$

which are necessary to derive the desired approximation estimates. Note that $\mathbf{D}^{p+1}(F \circ G_K) = 0$ because F is a polynomial of degree p .

³ The triangles in $\Delta_h(\Omega)$ are curved if the polynomial basis functions in $V_h(\hat{\Omega})$ have degree $p > 1$.

This knowledge of isoparametric finite elements is sufficient to tackle our initial problem: construct a finite dimensional counterpart of $V(\Omega^{(k)})$ to solve (23) on $\Omega^{(k)}$ (where $\Omega^{(k)} := T^{(k)}(\Omega^0)$).

In the first iteration, we construct $V_h(\Omega^0)$ in the isoparametric fashion described above. First, we generate a triangulation of a suitable polytope $\hat{\Omega}^0$ that approximates Ω^0 . Then, we define the finite element space $V_h(\hat{\Omega}^0)$ and generate a transformation $F^{(0)} \in (V_h(\hat{\Omega}^0))^d$ that maps $\hat{\Omega}^0$ onto Ω^0 . Finally, we construct $V_h(\Omega^0)$ by combining reference local basis functions with the diffeomorphism $F^{(0)}$.

In the next iteration, we construct $V_h(\Omega^{(1)})$ in the same way, but replacing the diffeomorphism $F^{(0)}$ with the interpolant

$$(V_h(\hat{\Omega}^0))^d \ni F^{(1)} := \mathcal{I}_h \left(T^{(1)} \circ F^{(0)} \right),$$

where \mathcal{I}_h denotes the interpolation operator onto $(V_h(\hat{\Omega}^0))^d$. Since

$$T^{(1)}(\mathbf{x}) = \mathbf{x} + dT^{(0)}(\mathbf{x}),$$

the map $F^{(1)}$ can be written as

$$F^{(1)} = F^{(0)} + \mathcal{I}_h \left(dT^{(0)} \circ F^{(0)} \right).$$

Repeating this procedure at every iteration results in the isoparametric space $V_h(\Omega^{(k)})$ being constructed with the map

$$\begin{aligned} F^{(k)} &= \mathcal{I}_h \left(T^{(k)} \circ F^{(0)} \right) \\ &= \mathcal{I}_h \left(T^{(k-1)} \circ F^{(0)} \right) + \mathcal{I}_h \left(dT^{(k-1)} \circ T^{(k-1)} \circ F^{(0)} \right) \\ (26) \quad &= F^{(k-1)} + \mathcal{I}_h \left(dT^{(k-1)} \circ F^{(k-1)} \right), \end{aligned}$$

where the second equality follows from $F^{(k-1)} = \mathcal{I}_h(T^{(k-1)} \circ F^{(0)})$.

In general, the map $F^{(k)}$ may not fulfill the condition (24). However, by $W^{1,\infty}$ -error estimates of \mathcal{I}_h [15, Thm. 4.3.4], it holds that

$$\det \left(\mathbf{D}F^{(k)} \right) (x) \rightarrow \det \left(\mathbf{D} \left(T^{(k)} \circ F^{(0)} \right) \right) (x) \quad \text{as } h \rightarrow 0.$$

This, in light of Theorem 3.2, guarantees that $F^{(k)}$ is indeed a diffeomorphism if h is small enough (because $T^{(k)} \circ F^{(0)}$ is a diffeomorphism as well, and therefore $\det(\mathbf{D}(T^{(k)} \circ F^{(0)}))(x) \neq 0$). Additionally, note that the element transformation $G_K : \hat{K} \rightarrow K$ is affine, and thus,

$$\mathbf{D}^\alpha \left(F^{(k)} \circ G_K \right) = \left(\mathbf{D}^\alpha \left(F^{(k)} \right) \circ G_K \right) (\mathbf{D}G_K)^\alpha.$$

Therefore,

$$(27) \quad \left\| \mathbf{D}^\alpha \left(F^{(k)} \circ G_K \right) \right\| \leq \left\| \left(\mathbf{D}^\alpha \left(F^{(k)} \right) \circ G_K \right) \right\| \|(\mathbf{D}G_K)^\alpha\| \leq \left\| \mathbf{D}^\alpha \left(F^{(k)} \right) \right\| h^\alpha.$$

The estimate (27) is asymptotically equivalent to (25). This implies that modifying the transformation used to generate the isoparametric finite element space does not affect its approximation properties as long as $\|\mathbf{D}^\alpha(F^{(k)})\|$ is moderate.

Remark 6. It is not strictly necessary to replace the transformation $T^{(k)} \circ F^{(0)}$ with its interpolant $F^{(k)}$. However, evaluating $T^{(k)}$ can be computationally expensive and may not be supported natively in finite element software (in particular, if evaluating $T^{(k)}$ involves a complicated formula). On the other hand, as explained in the section 3, $T^{(k)}$ lies in practice in a finite dimensional space. Therefore, the interpolation operator \mathcal{I}_h can be represented as a matrix, which can be used to significantly speed up the computation of the update $dT^{(k)}$, as explained in section 6.

Remark 7. Usually, the isoparametric transformation F is chosen to be the identity on elements of $\Delta_h(\Omega)$ that do not share edges/faces with $\partial\Omega$. This particular choice of F is made to decrease the computational cost of matrix/vector assembly and is not dictated by error analysis. In our approach, the function F will generally differ from the identity even in the interior of the domain.

Remark 8. In [23, 35], the authors suggest pursuing shape optimization in Lagrangian coordinates by reformulating shape optimization problems as an optimal control problem on the initial domain. The resulting method is formally equivalent to the one presented in this work, but implies hard-coding of geometric transformation into shape functionals and PDE-constraints (which is problem dependent) and requires the derivation of Fréchet derivatives in Lagrangian coordinates (which are usually not considered in the shape optimization literature). In contrast, the approach presented in this work exploits the fact that these geometric transformations are included in standard finite element software and allows the use of formulae for Fréchet derivatives that are already available in the literature.

6. Implementation aspects. The previous sections consider different discretization aspects of the shape optimization problem (2). Section 3 introduces the finite dimensional space Q_N to construct the sequence of diffeomorphisms (10). Section 5, on the other hand, introduces the finite dimensional space $(V(\Omega^0))^d$ to approximate the solution of (2b) by means of isoparametric finite elements.

There are conflicting demands on the choice of these two finite dimensional spaces. On the one hand, employing the same discretization based on piecewise linear Lagrangian finite elements greatly simplifies the implementation in existing finite element libraries and may reduce the execution time. On the other hand, a decoupled discretization facilitates enforcing stability in the optimization process. For instance, the authors of [3, 29] suggest the use of linear Lagrangian finite elements built on two nested meshes: a coarser one to discretize the geometry and a finer one to solve the state equation. They report that this reduces the presence of spurious oscillations in the optimized shape. The authors of [25] advocate a similar practice in the context of isogeometric analysis shape optimization.

A decoupled discretization may also be required if one aims at using higher-order approximations of the state variable u . To elucidate this, recall that the use of higher-order finite elements is motivated only if the exact solution is sufficiently regular. More specifically, isoparametric finite element solutions of degree p converge as $\mathcal{O}(h^p)$ in the energy norm provided that the exact solution satisfies $u \in H^{p+1}(\Omega)$, whereas the convergence rate deteriorates if u is less regular [15, Thm 3.2.1] or if Ω is not a polytope and isoparametric finite elements are replaced by standard affine-parametric finite elements [15, Rem. 4.4.4 (ii)].

It is virtually impossible to prescribe universal and sharp rules that ensure that the solution of the state constraint remains sufficiently regular during the optimization process, but elliptic regularity theory can provide some guidelines. Assuming the

problem data is sufficiently smooth, the solution of a linear elliptic Dirichlet BVP is H^s -regular [13, Def. 7.1] when Ω has a C^s -boundary [30, Thm. 8.13] (see also [32] for an extensive treatment of elliptic regularity theory). Therefore, it might be desirable to employ sufficiently regular transformations so that the regularity of the domain is preserved during the optimization process. In this case, typical isoparametric Lagrangian finite elements may not be a good choice for Q_N because they only allow $W^{1,\infty}$ piecewise polynomial representations of the domain transformations. The natural alternative is to employ multivariate B-splines of degree \tilde{p} [37, Def. 4.1], which are piecewise polynomials with compact support and are both $W^{\tilde{p},\infty}$ and $C^{\tilde{p}-1}$ -regular.

For these reasons, we focus on the more general case of a decoupled discretization of $(V(\Omega^0))^d$ and Q_N and discuss the implementation details for the following simple optimization algorithm, which covers all fundamental aspects of shape optimization.

Minimal shape optimization pseudocode.

1. Initialize; then, for $k \geq 0$:
2. compute the state u and evaluate J ; stop if converged; otherwise,
3. compute the update dT solving (18),
4. choose s such that $T + sdT \circ T$ is feasible and J is minimal,
5. update T and go back to step 2.

Step 1. First, we construct the finite element space $(V_h(\Omega^0))^d := \text{span}\{\mathbf{v}_i\}_{i=1}^M$ and store the coefficient vector $\mathbf{f}^{(0)} \in \mathbb{R}^M$ of the transformation $F^{(0)} \in (V_h(\Omega^0))^d$, which maps Ω^0 onto (an approximation of) Ω^0 . Then, we construct the space $Q_N := \text{span}\{\mathbf{q}_i\}_{i=1}^N$, initialize the vector field $T^{(0)} \in Q_N$ to the identity, and store its coefficients in the vector $\mathbf{t}^{(0)} \in \mathbb{R}^N$. Finally, we store the matrix representation \mathbf{I}_h of the interpolation operator

$$\mathcal{I}_h : Q_N \rightarrow (V_h(\Omega^0))^d.$$

The matrix \mathbf{I}_h is sparse if the basis functions $\{\mathbf{v}_i\}_{i=1}^M$ and $\{\mathbf{q}_i\}_{i=1}^N$ have (small) compact support.

Step 2. First, we compute the coefficients of $F^{(k)} := \mathcal{I}_h(T^{(k)} \circ F^{(0)})$. This is done by computing

$$(28) \quad \mathbf{f}^{(k)} = \mathbf{f}^{(k-1)} + \mathbf{I}_h \mathbf{d}\mathbf{t}^{(k-1)},$$

as will be justified in the next step. Then, we approximate $u_{\Omega^{(k)}}$ by means of isoparametric finite elements and evaluate \mathcal{J} on the domain $\Omega^{(k)}$. If the convergence criteria are not satisfied, we proceed further and compute an update of $T^{(k)}$.

Step 3. First, we have to assemble the load vector $\mathbf{d}\tilde{\mathbf{J}}_{\mathbf{q}}^{(k)} := \{dJ(T^{(k)}; \mathbf{q}_i)\}_{i=1}^N$. This can be computationally expensive because dJ depends on u , which is approximated with a finite element function and lives on a finite element mesh. Therefore, to evaluate dJ it is necessary to loop through the finite element mesh. Although one loop is sufficient if one evaluates the contribution of each cell for all basis functions in Q_N , evaluating these functions can be computationally expensive and may require extensive modifications of finite element software. Therefore, it may be desirable to employ a different strategy, which we detail below.

Let $\{\mathbf{v}_i^{(k)}\}_{i=1}^M$ denote the isoparametric basis of $(V_h(\Omega^{(k)}))^d$. The vector

$$(29) \quad \mathbf{d}\tilde{\mathbf{J}}_{\mathbf{v}^{(k)}}^{(k)} := \left\{ dJ\left(T^{(k)}; \mathbf{v}_i^{(k)}\right) \right\}_{i=1}^M$$

can be assembled efficiently with existing software, because the basis functions $\{\mathbf{v}_i^{(k)}\}$ are generally included in finite element software (while $\{\mathbf{q}_i\}$ may not be). Interestingly, the product of the transpose of the interpolation matrix \mathbf{I}_h^\top with (29) can be interpreted as the approximation

$$(30) \quad \mathbf{I}_h^\top \mathbf{d}\tilde{\mathbf{J}}_{\mathbf{v}^{(k)}}^{(k)} \approx \left\{ dJ \left(T^{(k)}; \mathbf{q}_i \circ \left(F^{(0)} \right)^{-1} \circ \left(T^{(k)} \right)^{-1} \right) \right\}_{i=1}^N,$$

where the right-hand side corresponds to the evaluation of $dJ(T^{(k)}; \cdot)$ on functions that move along with the domain transformation (see Figure 1). To explain the nature of the approximation in (30), we introduce a new finite dimensional space

$$(31) \quad Q_N^{(k)} := \text{span} \left\{ \mathbf{q}_i^{(k)} := \mathbf{q}_i \circ \left(F^{(0)} \right)^{-1} \circ \left(T^{(k)} \right)^{-1} \right\}_{i=1}^N.$$

The space $Q_N^{(k)}$ arises naturally if one considers shape optimization in Lagrange coordinates [23, 25, 35], and satisfies $Q_N^{(k)} \subset W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$. Compared to Q_N , this new space $Q_N^{(k)}$ has a great computational advantage: the previously computed matrix \mathbf{I}_h corresponds to the representation of the interpolation operators

$$(32) \quad \mathcal{I}_h : Q_N \rightarrow (V_h(\Omega^0))^d$$

as well as to the representation of

$$(33) \quad \mathcal{I}_h^{(k)} : Q_N^{(k)} \rightarrow (V_h(\Omega^{(k)}))^d$$

with respect to the basis $\{\mathbf{q}_i^{(k)}\}_{i=1}^N$ and $\{\mathbf{v}_i^{(k)}\}_{i=1}^M$. This implies that the vector $\mathbf{d}\tilde{\mathbf{J}}_{\mathbf{q}^{(k)}}^{(k)}$ (the right-hand side of (30)) can be assembled more easily and quickly than $\mathbf{d}\tilde{\mathbf{J}}_{\mathbf{q}}^{(k)}$. On top of that, (33) motivates the interpretation (30). Note that for given test cases, the asymptotic rate of convergence in this approximation can be explicitly computed.

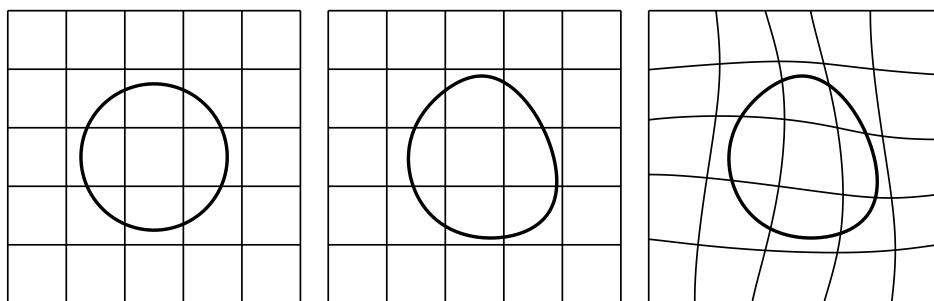


FIG. 1. Left: *initial configuration*. The initial domain Ω^0 is a circle, and the discretization Q_N is constructed defining its basis functions \mathbf{q}_i 's on a regular grid. To update the domain Ω^0 , one constructs a map $F^{(1)} \in Q_N$. Center: the domain Ω^0 is deformed into a perturbed circle according to $F^{(1)}$. To further perturb the domain, one retains the space Q_N and constructs a map $F^{(2)} \in Q_N$. Alternatively (right): after deforming the domain Ω^0 according to $F^{(1)}$, one defines a new discretization $Q_N^{(1)}$ by constructing its basis functions $\mathbf{q}_i^{(1)}$'s via pullback and uses this new discretization to compute the new update $F^{(2)}$.

Once the load vector for the update step has been assembled, we need to assemble the stiffness matrix. In principle, this should be done as well with respect to the new discretization (31). Alternatively, one can redefine the inner-product of \mathcal{X} so that

$$(\mathbf{q}_i, \mathbf{q}_j)_{\mathcal{X}} = \left(\mathbf{q}_i^{(k)}, \mathbf{q}_j^{(k)} \right)_{\mathcal{X}^{(k)}} \quad \text{for all } i, j = 1, \dots, N,$$

and the stiffness matrix can be computed once and for all in the initialization step.

Finally, one computes the vector $\mathbf{dt}^{(k)}$, which contains the coefficients of the series expansion of $dT^{(k)}$ with respect to the basis of (31). Computing $\mathbf{I}_h \mathbf{dt}^{(k)}$ returns the coefficients of the interpolant of $dT^{(k)}$ in $(V_h(\Omega^{(k)}))^d$. Due to the isoparametric nature of its basis $\{\mathbf{v}_i^{(k)}\}_{i=1}^M$, these coefficients equal those resulting from interpolating $dT^{(k)} \circ F^{(k)}$ onto $(V(\hat{\Omega}^0))^d$. This explains why (28) is the algebraic counterpart of (26).

Steps 4–5. Computing s is a one-dimensional optimization problem and can be solved (at least approximately) by line search. Since the focus of this work is on discretization of transformations, we have not investigated which line search algorithm is best suited to find s , and we defer this task to future research. We only mention that, in our experience, the strategies described in [45, sect. 3] generally lead to good results.

Independently on the choice of the line search algorithm, we have to evaluate J on $T^{(k)} + sdT^{(k)}$ for various s . In light of (28), this means computing the state variable u using isoparametric finite elements⁴ whose coefficients are

$$\mathbf{f}_s^{(k+1)} = \mathbf{f}^{(k)} + s\mathbf{I}_h \mathbf{dt}^{(k)}$$

and choosing s such that the value of J is minimal. Of course, one has to enforce admissibility of $T^{(k)} + sdT^{(k)}$. By Theorem 3.2, it is sufficient to verify that the minimum of

$$(34) \quad \det \left(\mathbf{D} \left(F^{(k)} + s\mathcal{I}_h \left(dT^{(k)} \right) \right) \right) \approx \det \left(\mathbf{D} \left(T^{(k)} + sdT^{(k)} \right) \right)$$

remains positive. However, note that small values of (34) may negatively affect the ellipticity constant of the BVP (2b), which in turn negatively affects the constant of finite element error estimates. Finally, one rescales $dT^{(k)} := sdT^{(k)}$, sets $T^{(k+1)} = (\mathcal{I} + dT^{(k)}) \circ (T^{(k)})$, and goes back to step 2.

7. Numerical experiments. We split our numerical investigations in two parts. In the first one, we consider a PDE-constrained shape optimization problem that admits stable minimizers. We use this test case to investigate the approximation properties of the algorithm presented in section 6 for different discretizations of control and state variables. In the second part, we test our approach on a more challenging shape optimization problem for which analytical solutions are unavailable, and we provide qualitative evidence that increasing the degree of the transformation can be beneficial. The numerical results are obtained with a code based on the finite element library Firedrake [4, 5, 7, 8, 16, 33, 42, 50].

7.1. Bernoulli free-boundary problem. As the first test case, we consider the Dirichlet BVP

$$(35) \quad -\Delta u = 0 \quad \text{in } \Omega, \quad u = u_{\text{in}} \quad \text{on } \partial\Omega^{\text{in}}, \quad u = u_{\text{out}} \quad \text{on } \partial\Omega^{\text{out}},$$

⁴Note that the same strategy can be employed with any type of finite elements that are constructed using a reference element.

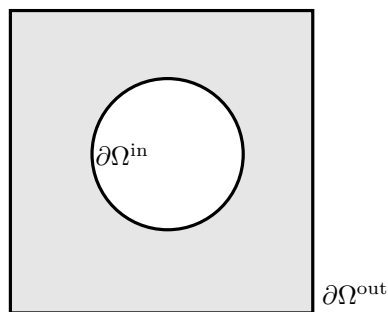


FIG. 2. Computational domain for the Dirichlet BVP (35). The external boundary $\partial\Omega^{\text{out}}$ is a square centered at the origin with a corner in $(1, 1)$. The internal boundary $\partial\Omega^{\text{in}}$ is optimized to achieve $\frac{\partial u}{\partial \mathbf{n}}|_{\partial\Omega^{\text{in}}} = g$.

stated on the domain $\Omega \subset \mathbb{R}^2$ depicted in Figure 2. The goal is to find the shape of $\partial\Omega^{\text{in}}$ so that the Neumann trace $\frac{\partial u}{\partial \mathbf{n}}$ is equal to a prescribed function g on $\partial\Omega^{\text{in}}$. For the sake of simplicity, we assume that the Dirichlet data u_{in} and the Neumann data g are constants and that only $\partial\Omega^{\text{in}}$ is free to move.

This Bernoulli free boundary problem can be reformulated as the following shape optimization problem [19]:

$$(36) \quad \inf_{\Omega \in \mathcal{U}_{\text{ad}}} \mathcal{J}(\Omega, u) = \int_{\Omega} \nabla u \cdot \nabla u + g^2 \, d\mathbf{x} \quad \text{subject to} \quad \begin{cases} -\Delta u = 0 & \text{in } \Omega, \\ u = u_{\text{in}} & \text{on } \partial\Omega^{\text{in}}, \\ u = u_{\text{out}} & \text{on } \partial\Omega^{\text{out}}, \end{cases}$$

whose Fréchet derivative reads [35]

$$(37) \quad dJ(T; \mathcal{T}) = \int_{T(\Omega_0)} \operatorname{div} \mathcal{T} (\nabla u_{T(\Omega_0)} \cdot \nabla u_{T(\Omega_0)} + g^2) - \nabla u_{T(\Omega_0)} \cdot (\mathbf{D}\mathcal{T} + \mathbf{D}\mathcal{T}^\top) \nabla u_{T(\Omega_0)} \, d\mathbf{x}.$$

In [19], the authors have studied this shape optimization problem (36) in detail and, performing shape analysis in polar coordinates, have shown that the shape Hessian is both continuous and coercive (when restricted to normal perturbations) in the $H^{1/2}(\partial\Omega)$ -norm. For this reason, minimizers of (36) are stable.

To construct a test case for our numerical simulations, we set the optimal shape of $\partial\Omega^{\text{in}}$ to be a circle centered at the origin with radius 0.4. For such a choice of $\partial\Omega^{\text{in}}$, the function (expressed in polar coordinates)

$$u(r, \varphi) = \ln(0.4) - \log(r)$$

satisfies the Dirichlet BVP (35) with $u_{\text{in}} = 0$ and $u_{\text{out}} = u$. The Neumann data on the interior boundary is $g = 2.5$. The value J_{\min} of the misfit functional in the optimal shape is approximately 28.30694161405. This value has been computed with quadratic isoparametric finite elements on a sequence of nested meshes; the relative error between the value of the misfit functional computed on the last and on the second last mesh is approximately $6 \cdot 10^{-11}$.

As an initial guess, we set $\partial\Omega_0^{\text{in}}$ to be a circle of radius 0.5 centered at $(0.04, 0.05)$. Note that we have repeated these numerical experiments with three other choices for

the initial guess $\partial\Omega_0^{\text{in}}$ and have obtained similar results. These alternative initial guesses are a circle of radius 0.47 centered at $(0.07, 0.03)$, a circle of radius 0.55 centered at $(-0.1, 0)$, and a circle of radius 0.5367 centered at $(-0.137, 0.03)$.

To discretize geometric transformations, we consider linear/quadratic/cubic tensorized Schoenberg B-splines constructed on regular grids [56]. These grids are refined uniformly (with widths ranging from 1.8×2^{-1} to 1.8×2^{-6}) and are contained in a square (the hold-all domain D) that is centered at the origin and has a corner at $(0.95, 0.95)$, so that $\partial\Omega^{\text{out}}$ is not modified in the optimization process. Finite element approximations of $u_{T(\Omega_0)}$ are computed with linear/quadratic isoparametric finite elements on a sequence of five triangular meshes generated using uniform refinement in Gmsh [28]. Note that finer meshes are adjusted to fit curved boundaries.

The optimization is carried out by repeating the following simple procedure for a fixed number of iterations: at every iteration, we compute a $H_0^1(D)$ -descent direction dT by solving (18) and choose the optimization step $s \in \{0, 0.1, \dots, 1\}$ that minimizes $J(T + sdT \circ T)$. Such a simple optimization strategy is sufficient for these numerical experiments, although we are aware that it is not efficient. The development of more efficient optimization strategies in the context of shape optimization is a current topic of research. In [55], the authors obtained promising results with a BFGS-type algorithm based on a Steklov–Poincaré metric. We defer to future research the numerical comparison of optimization strategies for shape optimization.

In Figure 3, we plot two steps of this simple optimization strategy. Transformations are discretized with quadratic B-splines built on the fourth grid, whereas the state $u_{T(\Omega_0)}$ is approximated with linear finite elements on the second coarsest mesh. Qualitatively, we observe the expected behavior of a (truncated) linesearch. The predicted-descent line is given by $J(T) - dJ(T, sdT)$, with $s = 0, 0.1, 0.2, 0.3$, and is tangential to $J(T + sdT \circ T)$ at $s = 0$. This shows empirically that formula (37) is correct.

Next, we investigate to which accuracy we can solve shape optimization problems. In particular, we study the impact of the discretization (polynomial-) degree and (refinement-) level used for the control and the state. This is done systematically by keeping certain discretization parameters fixed and varying the remaining ones. To simplify the exposition, we associate the term *grid* to the discretization of the control (that is, of geometric transformations), whereas the term *mesh* refers to the discretization of the state).

First, we fix the control discretization to quadratic B-splines built on the finest grid. For each finite element mesh (previously generated with Gmsh), we perform 101 steps of our simple optimization strategy employing linear finite element

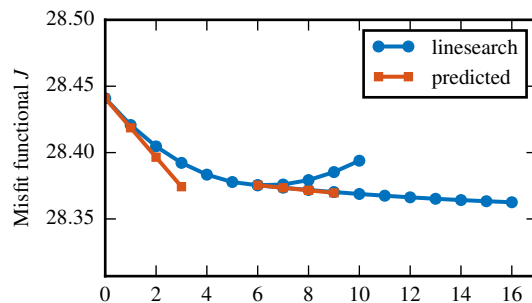


FIG. 3. Evolution of J on two optimization steps. The second linesearch starts at the minimum of the first one. The predicted descent is computed evaluating dJ on the (selected) descent direction.

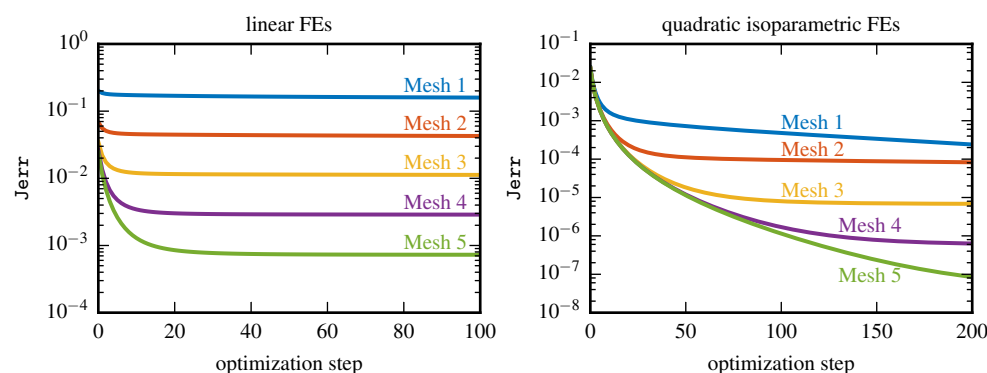


FIG. 4. Evolution of J_{err} when the state variable is discretized with linear (left) and quadratic (right) finite elements on different meshes constructed via uniform refinement. Convergence lines saturate at different levels, which decay algebraically with respect to the mesh width.

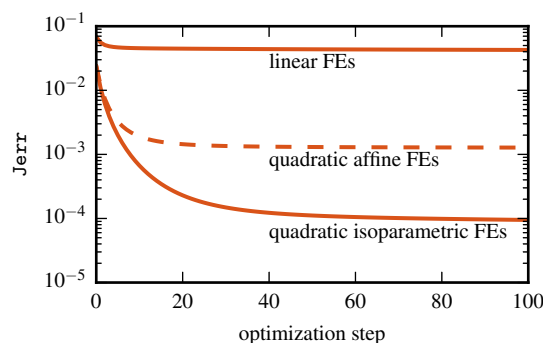


FIG. 5. Evolution of J_{err} when the state variable is discretized with linear, quadratic affine, and quadratic isoparametric finite elements on the second mesh. Quadratic affine finite elements perform worse than their isoparametric counterpart.

approximations of the state $u_{T(\Omega_0)}$. Although not displayed, the sequence of shapes always converges qualitatively to the optimum. For a quantitative comparison, we store the minimum of J for every linesearch and plot the absolute error with respect to J_{\min} in Figure 4 (we plot a convergence history for each mesh). Henceforth, we use the notation J_{err} to refer to this absolute error. We observe that the convergence history lines saturate at different levels. In particular, the saturation level decays algebraically with respect to the mesh width. To further investigate the impact of finite element approximations on shape optimization, we repeat this experiment with quadratic isoparametric finite elements. Again, we observe algebraic convergence with respect to the mesh width, but at higher convergence rate (note the difference in the y-axis scale). In order to reach the saturation level on finer meshes, more optimization steps have to be carried out. This issue has been observed in [35] as well and is probably due to the simplicity of the optimization strategy. Before proceeding further, let us point out that the saturation level worsens if quadratic isoparametric finite elements are replaced by quadratic affine finite elements; see Figure 5.

In the previous experiments, we kept the discretization of transformations fixed. Now, we test different discretizations. In Figure 6, we consider two different discretization degrees of the control variable (linear/cubic B-splines). For each of these discretization degrees, we consider six grids and five meshes. The state is approximated

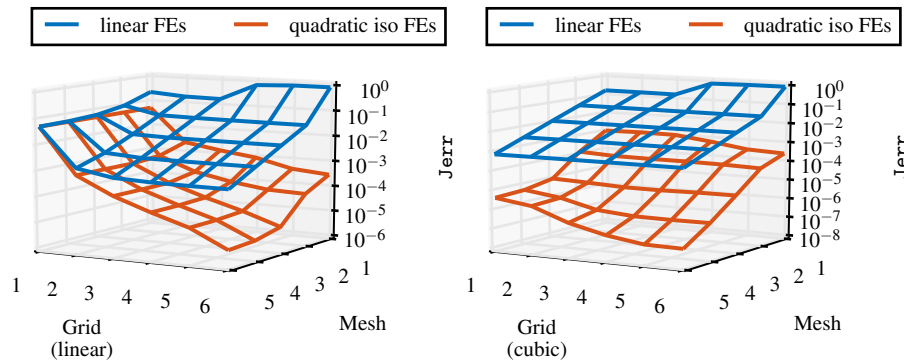


FIG. 6. Value of J_{err} after 201 optimization steps for different combination of control and state discretization. The algebraic rate of convergence with respect to mesh refinement and the benefits of higher-degree discretizations of transformations are clearly visible.

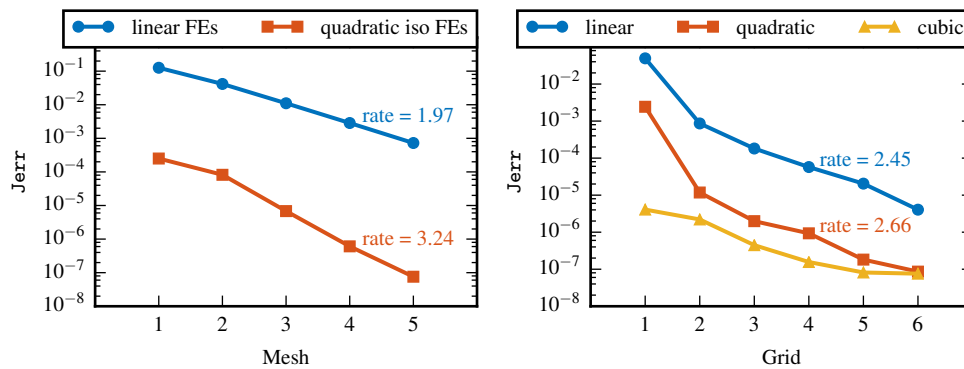


FIG. 7. Details from Figure 6. Left: convergence with respect to finite element discretization of the state. Right: convergence with respect to B-spline discretization of the control.

once with linear and once with quadratic isoparametric finite elements. For each combination, we plot J_{err} after 200 iterations. We observe that the discretization both of the control and of the state have an impact on J_{err} (the algebraic decay with respect to the finite element mesh width is conspicuous).

In Figure 7 (left), we consider the finest level and highest degree of the control discretization. We plot J_{err} (after 200 iterations) versus the finite element mesh index and consider both linear and quadratic isoparametric finite element approximations of the state. The algebraic rates of convergence for linear and quadratic finite elements read 1.97 and 3.24, respectively. These rates are in line with our expectations because duality techniques can be employed to prove superconvergence in the finite element approximation of the quadratic functional J [36]. Figure 6 shows also that J_{err} is almost entirely dominated by the error in the state when transformations are discretized with cubic B-splines. This is better highlighted in Figure 7 (right). There, we fix the state discretization to quadratic isoparametric finite elements on the finest mesh and plot J_{err} versus the grid index for linear, quadratic, and cubic B-splines. For instance, grid refinement for cubic B-splines has a very mild impact on J_{err} (due to the finite element approximation error of the state). We observe that the algebraic convergence rate for grid refinement of linear B-splines is approximately 2.45.

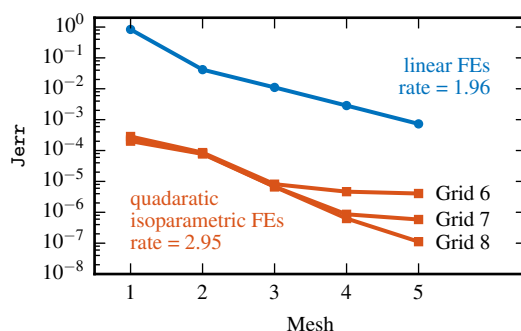


FIG. 8. Surprisingly, discretizing transformations with linear B -splines instead of cubic does not affect the rate of convergence of J_{err} with respect to finite element discretization of the state (compare with Figure 7, left).

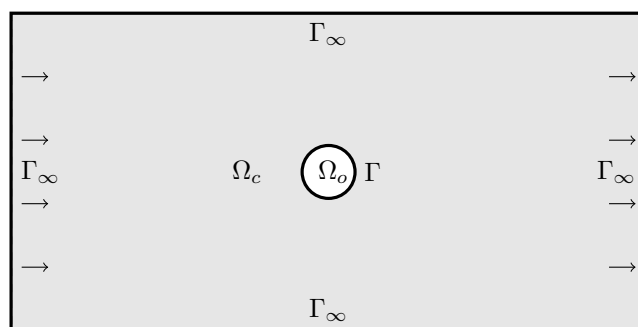


FIG. 9. Computational domain for the Stokes BVP (38). The internal boundary encloses the obstacle Ω_o to be shape optimized; the remaining boundary is a rectangle of width 12 and height 6.

This rate is higher than our expectations (a perturbation analysis via the Strang lemma would give rise to only linear convergence).

Finally, we consider the highest discretization level of the control and investigate whether its discretization degree has an impact on the rate of convergence of J_{err} with respect to the finite element discretization of the state. Let us recall that the regularity of the state on a perturbed domain depends, in principle, on the regularity of the domain. When perturbed with less smooth transformations, the resulting domain may not guarantee that the regularity of the state is preserved. This may have a negative impact on the finite element approximation. In Figure 8, we plot J_{err} versus finite element mesh refinement (for both linear and quadratic finite elements) when transformations are discretized with linear B -splines. When the state is approximated with quadratic isoparametric finite elements, the control discretization error is negligible only on much finer grids. However, it seems that the finite element convergence rate is not affected by the discretization degree of the control (in Figure 7, left, the control is discretized with cubic B -splines, and a similar convergence rate is observed).

7.2. Drag minimization in Stokes flow. As the second test case, we consider the flow of a viscous fluid Ω_c around a two-dimensional obstacle Ω_o (see Figure 9). The goal is to find the shape of Ω_o with minimum drag. This shape optimization problem is widely studied in the shape optimization literature [11, 49, 54].

To model the viscous fluid, we use Stokes' equations in weak form, which read, find the velocity $\mathbf{u} \in H^1(\Omega_c; \mathbb{R}^2)$ and the pressure $p \in L_0^2(\Omega_c) = \{p \in L^2(\Omega_c) : \int_{\Omega_c} p \, d\mathbf{x} = 0\}$ of the fluid such that $\mathbf{u}|_{\Gamma_\infty} = \mathbf{u}_\infty$, $u|_\Gamma = 0$, and

$$(38) \quad \int_{\Omega_c} \sum_{i=1}^2 (\nabla u_i \cdot \nabla v_i) - p \operatorname{div} \mathbf{v} + q \operatorname{div} \mathbf{u} \, d\mathbf{x} = 0$$

for all $\mathbf{v} \in H^1(\Omega_c; \mathbb{R}^2)$ such that $\mathbf{v}|_{\Gamma_{\text{wall}}} = 0$ and $\mathbf{v}|_\Gamma = 0$ and all $q \in L_0^2(\Omega_c)$. It is known that (38) admits a unique solution if the computational domain Ω_c is Lipschitz [31, 58].

Searching for the shape that minimizes the drag on the obstacle is equivalent to searching for the shape that minimizes the energy dissipated in the fluid due to shear forces [11]. The advantage of the latter is that the problem is self-adjoint. We model this dissipated energy with

$$(39) \quad \mathcal{J}(\Omega_c, \mathbf{u}, p) = \int_{\Omega_c} \sum_{i=1}^2 \nabla u_i \cdot \nabla u_i \, d\mathbf{x} \equiv \int_{\Omega_c} \|\mathbf{D}\mathbf{u}\|_F^2 \, d\mathbf{x}.$$

Its shape derivative reads⁵

$$(40) \quad d\mathcal{J}(T, \mathcal{T}) = \int_{T(\Omega_c)} \sum_{i=1}^2 (\nabla u_i \cdot \nabla u_i \operatorname{div} \mathcal{T} - 2 \nabla u_i \cdot \mathbf{D}\mathcal{T} \nabla u_i) \\ + 2 \operatorname{tr}(\mathbf{D}\mathbf{u} \mathbf{D}\mathcal{T}) p - 2p \operatorname{div} \mathbf{u} \operatorname{div} \mathcal{T} \, d\mathbf{x}.$$

Minimizing (39) subject to (38) alone is not interesting because energy dissipation can be reduced by shrinking or removing the obstacle. To obtain a more reasonable test case, we introduce two additional constraints: the area and the barycenter of the obstacle should remain constant. Similarly to [54], we model these constraints with the functionals

$$(41) \quad \mathcal{A}(T) := \int_{T(\Omega_c)} 1 \, d\mathbf{x} - \int_{\Omega_c} 1 \, d\mathbf{x}, \\ \mathcal{B}_i(T) := \int_{T(\Omega_c)} x_i \, d\mathbf{x} - \int_{\Omega_c} x_i \, d\mathbf{x} \quad \text{for } i = 1, 2.$$

For convenience, we enforce the constraints $\mathcal{A}(T) = 0$ and $\mathcal{B}_i(T) = 0$, $i = 1, 2$, onto the shape optimization problem in the form of penalty functions; that is, we replace the functional (39) with

$$(42) \quad \mathcal{J}_p(T) = \mathcal{J}(T) + \frac{\mu_0}{2} \mathcal{A}^2(T) + \sum_{i=1}^2 \frac{\mu_i}{2} \mathcal{B}_i^2(T),$$

where $\mathcal{A}^2(T) := (\mathcal{A}(T))^2$, $\mathcal{B}_i^2(T) := (\mathcal{B}_i(T))^2$, and $0 \leq \mu_i \in \mathbb{R}$, $i = 0, 1, 2$. The shape derivatives of the squared constraints are given by

$$(43) \quad d\mathcal{A}^2(T, \mathcal{T}) = 2\mathcal{A}(T) \int_{T(\Omega_c)} \operatorname{div} \mathcal{T} \, d\mathbf{x}, \\ d\mathcal{B}_i^2(T, \mathcal{T}) = 2\mathcal{B}_i(T) \int_{T(\Omega_c)} \operatorname{div}(x_i \mathcal{T}) \, d\mathbf{x} \quad \text{for } i = 1, 2.$$

⁵We believe that this volume based formula (which can be computed with [52]) is already known to the shape optimization community, although we did not manage to find it explicitly in available publications.

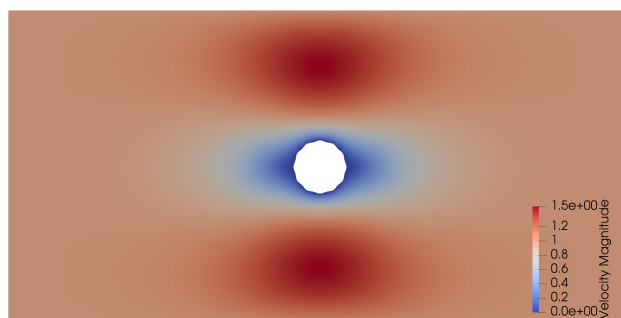


FIG. 10. Magnitude of \mathbf{u} when Ω_0 is a golf-ball-like shape centered at the origin with radius 0.5.

To investigate the effect of the degree of the discretization Q_N , we keep the discretization of the state equation fixed and visually compare optimized shapes obtained with linear/quadratic tensorized Schoenberg B-splines constructed on regular grids [56] of different widths. The penalty parameters are set to $\mu_0 = \mu_1 = \mu_2 = 100$.

To approximate (\mathbf{u}, p) , we use Taylor–Hood P2-P1 finite elements (which are affine-parametric) on a sufficiently fine triangular mesh (12858 nodes and 25264 triangles). This discretization is stable [18]. The resulting linear system can then be solved using GMRES and a block-diagonal preconditioner based on the stiffness matrix and on the mass matrix for the velocity- and the pressure-block, respectively [18].

As an initial guess, we set Ω_0 to be a golf-ball-like shape centered at the origin with radius 0.5. Figure 10 shows the magnitude of the velocity field \mathbf{u} for this initial guess. We use a golf ball (and not a disc) as an initial guess so that the transformation that maps the initial guess to the optimal shape cannot be obtained with a coarse discretization. Note that, although not analytically known, there is strong evidence that the optimal shape has the form of an ellipse (with two sharp kinks at its extremities) [49].

To discretize geometric transformations, we consider different uniformly refined grids (contained in the outer boundary of Ω_c) and construct the Schoenberg spline spaces. The grid width ranges from 6×2^{-1} to 6×2^{-4} . Note that the width of the outer boundary of Ω_c is twice its heights.

Instead of the simple optimization algorithm used in subsection 7.1, here we employ the optimization library ROL [51], which implements an L-BFGS algorithm combined with a linesearch that enforces the strong Wolfe conditions. The use of this more sophisticated optimization strategy is not strictly necessary, but it decreases greatly the number of PDE-solves.

To compute steepest descent directions, we first compute a Riesz-representative $\nabla \mathcal{J}$ of $d\mathcal{J}$ (in the space Q_N) with respect to the inner-product⁶ [38]

$$(44) \quad (\mathbf{p}, \mathbf{q})_{\text{CR}} := \int_{\Omega_c} \sum_{i=1}^2 (\nabla p_i \cdot \nabla q_i) + \mu \mathcal{B} \mathbf{p} \cdot \mathcal{B} \mathbf{q} \, d\mathbf{x},$$

where $\mu : \Omega_c \rightarrow (0, \infty)$ is a weighting function and $\mathcal{B} \mathbf{p} = (\partial_x p_1 - \partial_2 p_2, \partial_y p_1 + \partial_x p_2)^\top$ is the operator corresponding to the Cauchy–Riemann equations. Then, we update

⁶This inner-product is a modification (based on the Cauchy–Riemann equations) of the inner-product associated to linear elasticity. It has the advantage of producing transformation updates that deliver meshes with very good mesh quality. For more details, we refer to [38].

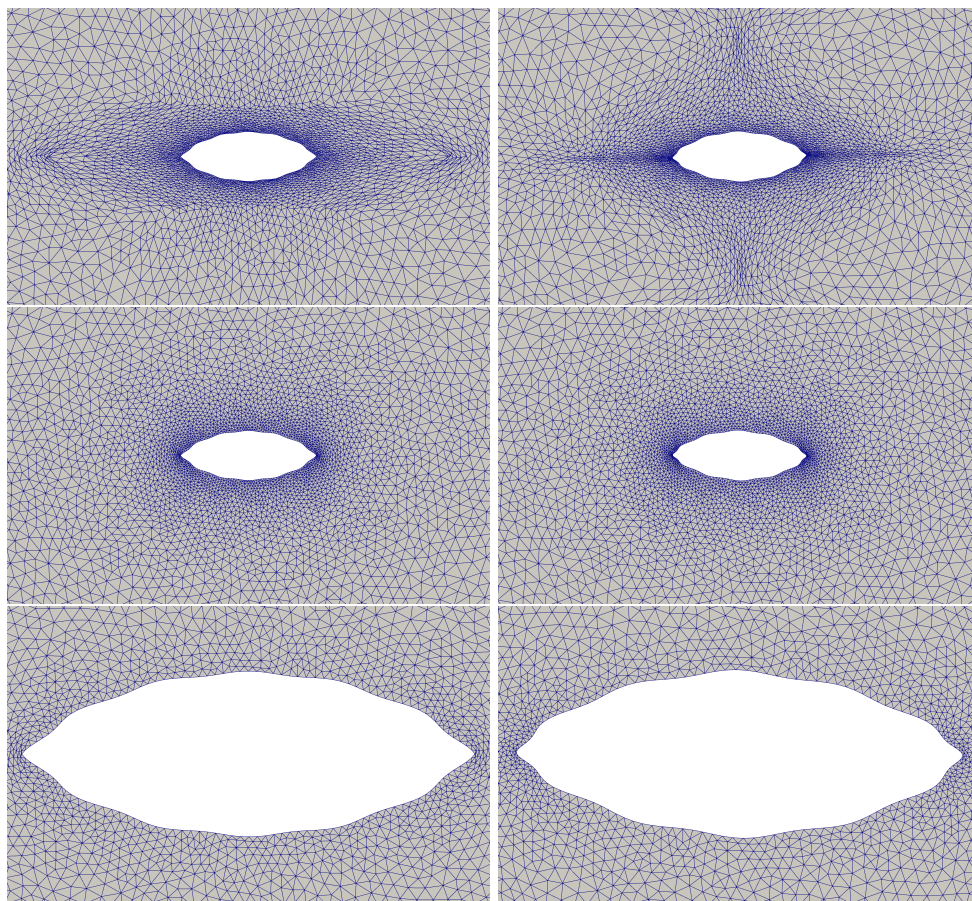


FIG. 11. Meshes obtained after 30 iterations of L-BFGS without (first row) and with (second row) the smoothing step (45). The space Q_N consists of affine (left, $N = 42$) and quadratic (right, $N = 64$) B-splines. We see that the smoothing step (45) compensates for the artifacts due to the coarseness of the discretization space Q_N . The third row is a close up of the second row.

$\nabla \mathcal{J}$ to $\nabla \mathcal{J}^{\text{BFGS}}$ using the L-BFGS two-loop recursion. Finally, we compute the mesh update $\mathbf{I}_h \nabla \mathcal{J}^{\text{BFGS}} \in (V_h(\hat{\Omega}^0))^d$ (see (28)). This mesh update is further regularized by computing the solution $\mathbf{df} \in (V_h(\hat{\Omega}^0))^d$ to

$$(45) \quad (\mathbf{df}, \mathbf{v})_{\text{CR}} = 0 \quad \text{for all } \mathbf{v} \in (V_h(\hat{\Omega}^0))^d, \quad \mathbf{df} = \mathbf{I}_h \nabla \mathcal{J}^{\text{BFGS}} \quad \text{on } \Gamma.$$

This further regularization is necessary when B-spline grids are very coarse because the space Q_N is not large enough to deform the shape optimally while keeping the mesh quality high; see Figure 11.

Figure 12 displays the optimized shapes after 100 L-BFGS iterations. Each row corresponds to a spline grid, whereas the left and the right columns contain shapes obtained using linear and quadratic splines, respectively. Note that, on the same grid, the space of quadratic splines contains slightly more elements (indicated by N) than the space with linear splines.

First, we observe that we recover the same optimal shape if N is sufficiently large (last row). Second, we observe that, on the same grid, a discretization based on quadratic splines results in a shape that is significantly closer to the optimal shape.

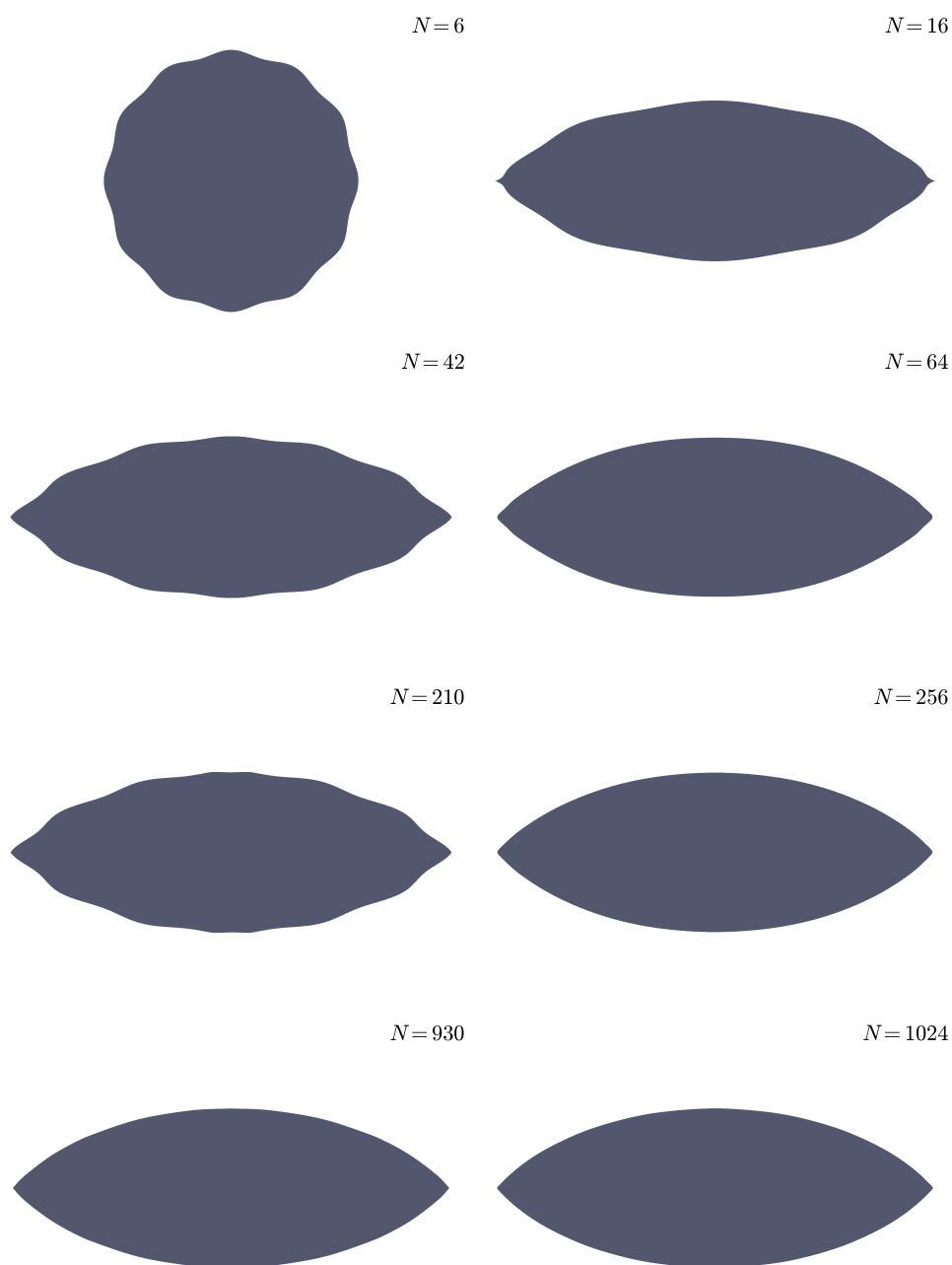


FIG. 12. Optimal shapes obtained (after 100 L -BFGS iterations) using affine (left column) or quadratic B-splines (right column) on grids of increasing resolution. The variable N denotes the number of B-spline basis-functions used to discretize transformations. We observe that, in general, the same qualitative result can be obtained using fewer quadratic basis functions.

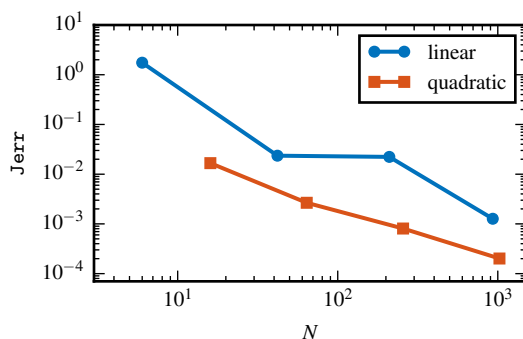


FIG. 13. Convergence as the grid is refined and the number of B-spline basis functions increases.

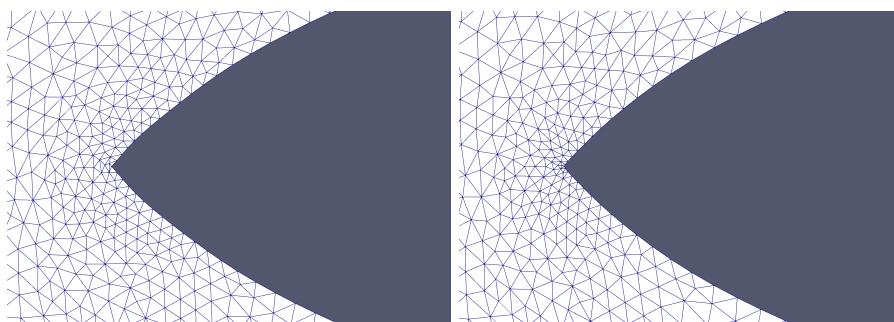


FIG. 14. Tip of the obtained shape using $N = 930$ affine B-splines (left) and $N = 1024$ quadratic B-splines.

Moreover, we see that the quadratic spline space needs many fewer basis functions to recover (qualitatively) the optimal shape. For instance, 64 quadratic B-splines yield a better shape than 210 linear B-splines.

For a quantitative comparison, we first optimize a shape using 4096 quadratic B-splines and 400 L-BFGS iterations and compute a reference optimal functional value \mathcal{J}^* (without penalty terms). Then, in Figure 13 we plot the value \mathbf{Jerr} obtained by subtracting \mathcal{J}^* from the value of \mathcal{J} on the optimized shapes displayed in Figure 12 (note that this is not an absolute difference). As in Figure 7 (right), we observe a similar rate of convergence for linear and quadratic B-splines. However, the quadratic B-spline error-line is at least one order of magnitude smaller than its linear counterpart.

Finally, we comment on the generation of sharp kinks at the extremities of the optimized shapes (which are predicted in [49]). Figure 14 shows the finite element mesh at the front tip of the optimal shapes obtained with affine and quadratic B-splines on the finest grid. These meshes have been obtained applying the optimized transformations to the mesh of the initial guess. We see that the tip of the shape obtained with quadratic B-splines is slightly rounded. This is because quadratic B-splines are continuously differentiable and cannot introduce the nonsmoothness that is required to generate the kink. However, note that the presence of the kink as a minor impact on the value of \mathcal{J} and that the shape obtained with quadratic B-splines (that lacks a sharp kink) has a smaller value of \mathcal{J} than the shape obtained with linear B-splines (see Figure 13).

8. Conclusion. We have formulated shape optimization problems in terms of deformation diffeomorphisms. This perspective simplifies the treatment of PDE-constrained shape optimization problems because it couples naturally with isoparametric finite element discretization of the PDE-constraint. In particular, it retains the asymptotic behavior of higher-order finite element discretization, and it allows the solution of PDE-constrained shape optimization problems to high accuracy, as confirmed by detailed numerical experiments. This shape optimization method can be implemented in standard finite element software and used to tackle challenging shape optimization problems that stem from industrial applications. The approach advocated is modular and can be combined with more advanced optimization algorithms, such as that of Schulz, Siebenborn, and Welker [55]; research in this vein will form the basis of future work.

REFERENCES

- [1] G. ALLAIRE, *Conception Optimale de Structures*, Springer-Verlag, Berlin, 2007.
- [2] G. ALLAIRE, F. JOUVE, AND A.-M. TOADER, *A level-set method for shape optimization*, C. R. Math. Acad. Sci. Paris, 334 (2002), pp. 1125–1130.
- [3] G. ALLAIRE AND O. PANTZ, *Structural optimization with FreeFem++*, Struct. Multidiscip. Optim., 32 (2006), pp. 173–181.
- [4] P. R. AMESTOY, I. S. DUFF, J.-Y. L'EXCELLENT, AND J. KOSTER, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 15–41.
- [5] P. R. AMESTOY, A. GUERMOUCHE, J.-Y. L'EXCELLENT, AND S. PRALET, *Hybrid scheduling for the parallel solution of linear systems*, Parallel Comput., 32 (2006), pp. 136–156.
- [6] P. F. ANTONIETTI, N. BIGONI, AND M. VERANI, *Mimetic Finite Difference Method for Shape Optimization Problems*, Springer International Publishing, Cham, Switzerland, 2015, pp. 125–132.
- [7] S. BALAY, S. ABHYANKAR, M. F. ADAMS, J. BROWN, P. BRUNE, K. BUSCHELMAN, L. DALCIN, V. ELJKHOUT, W. D. GROPP, D. KAUSHIK, M. G. KNEPLEY, L. C. MCINNES, K. RUPP, B. F. SMITH, S. ZAMPINI, H. ZHANG, AND H. ZHANG, *PETSc Users Manual*, Technical report ANL-95/11 - Revision 3.7, Argonne National Laboratory, Lemont, IL, 2016.
- [8] S. BALAY, W. D. GROPP, L. C. MCINNES, AND B. F. SMITH, *Efficient management of parallelism in object oriented numerical software libraries*, in Modern Software Tools in Scientific Computing, E. Arge, A. M. Bruaset, and H. P. Langtangen, eds., Birkhäuser Press, Boston, 1997, pp. 163–202.
- [9] K. BANDARA, F. CIRAK, G. OF, O. STEINBACH, AND J. ZAPLETAL, *Boundary element based multiresolution shape optimisation in electrostatics*, J. Comput. Phys., 297 (2015), pp. 584–598.
- [10] A. BARTON AND S. MAYBORODA, *Boundary-value problems for higher-order elliptic equations in non-smooth domains*, in Concrete Operators, Spectral Theory, Operators in Harmonic Analysis and Approximation, Springer, Basel, Switzerland, 2014, pp. 53–93.
- [11] J. A. BELLO, E. FERNÁNDEZ-CARA, J. LEMOINE, AND J. SIMON, *The differentiability of the drag with respect to the variations of a Lipschitz domain in a Navier–Stokes flow*, SIAM J. Control. Optim., 35 (1997), pp. 626–640.
- [12] M. BERGGREN, *A unified discrete-continuous sensitivity analysis method for shape optimization*, in Applied and Numerical Partial Differential Equations, Comput. Methods Appl. Sci. 15, Springer, New York, 2010, pp. 25–39.
- [13] D. BRAESS, *Finite Elements*, 3rd ed., Cambridge University Press, Cambridge, 2007.
- [14] C. J. BUDD, W. HUANG, AND R. D. RUSSELL, *Adaptivity with moving grids*, Acta Numer., 18 (2009), pp. 111–241.
- [15] P. G. CIARLET, *The Finite Element Method for Elliptic Problems*, SIAM, Philadelphia, 2002.
- [16] L. D. DALCIN, R. R. PAZ, P. A. KLER, AND A. COSIMO, *Parallel distributed computing using Python*, Adv. Water Res., 34 (2011), pp. 1124–1139.
- [17] M. C. DELFOUR AND J.-P. ZOLÉSIO, *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization*, 2nd ed., SIAM, Philadelphia, 2011.
- [18] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, 2014.

- [19] K. EPPLER AND H. HARBRECHT, *Efficient treatment of stationary free boundary problems*, Appl. Numer. Math., 56 (2006), pp. 1326–1339.
- [20] K. EPPLER AND H. HARBRECHT, *Shape optimization for free boundary problems—analysis and numerics*, in Constrained Optimization and Optimal Control for Partial Differential Equations, Internat. Ser. Numer. Math. 160, Birkhäuser/Springer, Basel, 2012, pp. 277–288.
- [21] L. C. EVANS, *Partial Differential Equations*, 2nd ed., American Mathematical Society, Providence, RI, 2010.
- [22] P. E. FARRELL, D. A. HAM, S. W. FUNKE, AND M. E. ROGNES, *Automated derivation of the adjoint of high-level transient finite element programs*, SIAM J. Sci. Comput., 35 (2013), pp. C369–C393.
- [23] D. FUEDER, B. SIMEON, AND A.-V. VUONG, *Fundamental aspects of shape optimization in the context of isogeometric analysis*, Comput. Methods Appl. Mech. Engrg., 286 (2015), pp. 313–331.
- [24] I. FUMAGALLI, N. PAROLINI, AND M. VERANI, *Shape optimization for Stokes flows: A finite element convergence analysis*, ESAIM Math. Model. Numer. Anal., 49 (2015), pp. 921–951.
- [25] D. FUSSEDER AND B. SIMEON, *Algorithmic Aspects of Isogeometric Shape Optimization*, Springer International Publishing, Cham, Switzerland, 2015, pp. 183–207.
- [26] M. GALEWSKI AND M. RADULESCU, *A Note on a Global Invertibility of Locally Lipschitz Functions on R^n* , ArXiv e-print, <https://arxiv.org/abs/1509.02965>, 2015.
- [27] H. GARCKE, C. HECHT, M. HINZE, C. KAHLE, AND K. F. LAM, *Shape optimization for surface functionals in Navier-Stokes flow using a phase field approach*, Interfaces Free Bound., 18 (2016), pp. 219–261.
- [28] C. GEUZAIN AND J.-F. REMACLE, *GMSH: A 3-d finite element mesh generator with built-in pre-and post-processing facilities*, Internat. J. Numer. Methods Engrg., 79 (2009), pp. 1309–1331.
- [29] M. GIACOMINI, O. PANTZ, AND K. TRABELSI, *Certified descent algorithm for shape optimization driven by fully-computable a posteriori error estimators*, ESAIM Control Optim. Calc. Var., 23 (2017), pp. 977–1001.
- [30] D. GILBARG AND N. S. TRUDINGER, *Elliptic Partial Differential Equations of Second Order*, Classics Math., Springer-Verlag, Berlin, 2001.
- [31] V. GIRAULT AND P.-A. RAVIART, *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms*, Springer Ser. Comput. Math., 5, Springer, Cham, 1986.
- [32] P. GRISVARD, *Elliptic Problems in Nonsmooth Domains*, Classics Appl. Math., 69, SIAM, Philadelphia, 2011.
- [33] B. HENDRICKSON AND R. LELAND, *A multilevel algorithm for partitioning graphs*, in Proceedings of the 1995 ACM/IEEE Conference on Supercomputing, San Diego, CA, 1995, p. 28.
- [34] M. HINZE, R. PINNAU, M. ULBRICH, AND S. ULBRICH, *Optimization with PDE Constraints*, Springer, New York, 2009.
- [35] R. HIPTMAIR AND A. PAGANINI, *Shape optimization by pursuing diffeomorphisms*, Comput. Methods Appl. Math., 15 (2015), pp. 291–305.
- [36] R. HIPTMAIR, A. PAGANINI, AND S. SARGHEINI, *Comparison of approximate shape gradients*, BIT, 55 (2015), pp. 459–485.
- [37] K. HÖLLIG, *Finite Element Methods with B-Splines*, SIAM, Philadelphia, 2003.
- [38] J. A. IGLESIAS, K. STURM, AND F. WECHSUNG, *Shape Optimisation with Nearly Conformal Transformations*, ArXiv e-print, <https://arxiv.org/abs/1710.06496>, 2017.
- [39] R. C. JAMES, *Characterizations of reflexivity*, Studia Math., 23 (1963/1964), pp. 205–216.
- [40] G. KATRIEL, *Mountain pass theorems and global homeomorphism theorems*, Ann. Inst. H. Poincaré Anal. Non Linéaire, 11 (1994), pp. 189–209.
- [41] A. LAURAIN AND K. STURM, *Distributed shape derivative via averaged adjoint method and applications*, ESAIM Math. Model. Numer. Anal., 50 (2016), pp. 1241–1267.
- [42] F. LUPORINI, D. A. HAM, AND P. H. J. KELLY, *An algorithm for the optimization of finite element integration loops*, ACM Trans. Math. Software, 44 (2017), pp. 3:1–3:26.
- [43] P. MORICE, *Une méthode d'optimisation de forme de domaine*, in Control Theory, Numerical Methods and Computer Systems Modelling, Springer, Berlin, 1975.
- [44] F. MURAT AND J. SIMON, *Etude de problèmes d'optimal design*, Springer, Berlin, 1976, pp. 54–62.
- [45] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, 2nd ed., Springer Ser. Oper. Res. Financ. Eng., Springer, New York, 2006.
- [46] A. PAGANINI, *Approximate shape gradients for interface problems*, in New Trends in Shape Optimization, Internat. Ser. Numer. Math. 166, Springer, Birkhäuser, Cham, Switzerland, 2015, pp. 217–227.
- [47] A. PAGANINI, *Numerical Shape Optimization with Finite Elements*, Ph.D. thesis, ETH, Zürich/Zürich, Switzerland, 2016.

- [48] A. PAGANINI AND R. HIPTMAIR, *Approximate Riesz representatives of shape gradients*, in Proceedings of the Conference on System Modeling and Optimization, 2015, pp. 399–409.
- [49] O. PIRONNEAU, *On optimum design in fluid mechanics*, J. Fluid. Mech., 64 (1974), pp. 97–110.
- [50] F. RATHGEBER, D. A. HAM, L. MITCHELL, M. LANGE, F. LUPORINI, A. T. T. MCRAE, G.-T. BERCEA, G. R. MARKALL, AND P. H. J. KELLY, *Firedrake: Automating the finite element method by composing abstractions*, ACM Trans. Math. Software, 43 (2016), pp. 24:1–24:27.
- [51] D. RIDZAL AND D. P. KOURI, *Rapid Optimization Library*, Tech. report, Sandia National Laboratories, Albuquerque, NM, 2014.
- [52] S. SCHMIDT, *Weak and strong form shape Hessians and their automatic generation*, SIAM J. Sci. Comp., 40 (2018), pp. C210–C233.
- [53] S. SCHMIDT, C. ILIC, V. SCHULZ, AND N. R. GAUGER, *Three-dimensional large-scale aerodynamic shape optimization based on shape calculus*, Amer. Inst. Aeronautics Astronautics J., 51 (2013), pp. 2615–2627.
- [54] V. SCHULZ AND M. SIEBENBORN, *Computational comparison of surface metrics for PDE constrained shape optimization*, Comput. Methods Appl. Math., 16 (2016), pp. 485–496.
- [55] V. H. SCHULZ, M. SIEBENBORN, AND K. WELKER, *Efficient PDE constrained shape optimization based on Steklov–Poincaré-type metrics*, SIAM J. Optim., 26 (2016), pp. 2800–2819.
- [56] L. L. SCHUMAKER, *Spline Functions: Computational Methods*, SIAM, Philadelphia, 2015.
- [57] K. STURM, *Convergence of Newton's Method in Shape Optimisation via Approximate Normal Functions*, ArXiv e-print, <https://arxiv.org/abs/1608.02699>, 2016.
- [58] R. TEMAM, *Navier-Stokes Equations: Theory and Numerical Analysis*, American Mathematical Society, Providence, RI, 2001.