

# IMPROVED DISCRETE BOUNDARY TYPE SHAPE GRADIENTS FOR PDE-CONSTRAINED SHAPE OPTIMIZATION\*

WEI GONG <sup>†</sup>, JIAJIE LI <sup>‡</sup>, AND SHENGFENG ZHU <sup>§</sup>

**Abstract.** We propose in this paper two kinds of continuity preserving discrete shape gradients of boundary type for PDE-constrained shape optimizations. First, a modified boundary shape gradient formula for shape optimization problems governed by elliptic Dirichlet problems was proposed recently based on the discrete variational outward normal derivatives. The advantages of this new formula over the previous one lie in the improved numerical accuracy and the continuity along the boundary. In the current paper we generalize this new formula to other shape optimization problems including the Laplace and Stokes eigenvalue optimization problems, the shape optimization of Stokes or Navier-Stokes flows and the interface identification problems. We verify this new formula's numerical accuracy in different shape optimization problems and investigate its performance in several popular shape optimization algorithms. The second contribution of this paper is to propose a continuous discrete shape gradients of boundary type for Neumann problems, by using the ideas of gradient recovery techniques. The continuity property of the discrete boundary shape gradient is helpful in certain shape optimization algorithms and provides certain flexibility compared to the previous discontinuous ones, which are extensively discussed in the current paper.

Keywords: shape optimization, shape gradient, boundary formulation, boundary correction, gradient recovery, finite element method

**1. Introduction.** Shape optimization is an old research field that can be traced back to the work of Hadamard in 1908 [18], and followed subsequently by many researchers from the perspectives of theoretical analysis (cf. [1, 13] and numerical simulations (cf. [10, 19, 20])). Nowadays, shape optimization is a hot research topic due to its increasingly wide applications in various fields of modern science and engineering [1, 15, 20, 37, 41]. As most of shape design problems involve partial differential equations (PDEs), it is difficult and usually impossible to obtain the optimal shapes from theoretical analysis. Thus, numerical techniques find their powers to seek “approximate” optimal shapes with the help of the efficient PDEs discretization methods and modern optimization algorithms [22].

For discretizations of PDEs in shape optimization problems, finite element method is among the most popular approaches (see e.g., [21]), partially due to its flexibility to deal with arbitrary computational domains that is essential in shape optimization algorithms because the computational domain usually changes after each iteration. For the shape evolution algorithm, gradient-type optimization methods are widely used to obtain the approximate optimal domain shapes. For other optimization algorithms, e.g., the Newton or quasi-Newton type optimization algorithms, we refer to [38, 40] and the references therein. To perform a gradient-type algorithm in shape optimization, an essential ingredient is the so-called *shape gradient*, which is used to obtain feasible descent directions and can be derived from *Eulerian derivative*. It is well recognized in the community of shape optimization that the Eulerian derivatives can be derived by *shape sensitivity analysis* [15, 41], which measures the variations of a certain shape functional with respect to shape variations of domain.

It is well-known that the Eulerian derivative of the shape functional has two representations, i.e., the distributed form and the boundary form, which are equivalent to each other under certain smoothness assumptions. The famous Hadamard-Zolésio structure theorem [15] shows that the shape gradient for a general shape functional can be obtained by computing the Eulerian derivative in the form of a boundary integral assuming that the domain is smooth enough. Several implications of this structure theorem make the boundary formula popular in the shape optimization community. First, it tells us that the directional derivative of the objective functional depends only on the normal component of the given direction on the boundary. Second, it gives an explicit choice of descent direction for the objective functional which can then be utilized in shape gradient descent algorithms. On the other hand, to obtain the descent direction by using the distributed shape gradient we have to solve an additional equation posed on the whole computational domain (cf. [28]), makes this approach costly in applications as the solutions to the state equation are already time-consuming. However, the distributed shape gradients do have advantages. On the one hand, the derivation of distributed one allows for less regular solutions and domains. On the other hand, the discretize-then-optimize and optimize-then-discretize approaches commute for the distributed shape gradient while not for the boundary shape gradient ([6]). Moreover, it is very well suited to certain shape optimization algorithms, for instance, the level set method [31] (see, e.g., [28] for solving a level-set transport equation based on a gradient flow associated with the distributed Eulerian derivative). For the use of distributed shape gradients in numerical algorithms, we refer to [8, 28, 39] and the references therein.

Recently, shape gradient of distributed type attracts more attentions in the shape optimization community,

---

\*The first author was supported in part by the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDB 41000000), the National Key Basic Research Program (Grant No. 2018YFB0704304) and the NSFC (Grant No. 12071468 and 11671391). The second and third authors were supported in part by Science and Technology Commission of Shanghai Municipality (No. 18dz2271000, No. 21JC1402500, and No. 19ZR1414100), and the NSFC under grant 12071149.

<sup>†</sup>LSEC, Institute of Computational Mathematics & NCMIS, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, 100190 Beijing, China. E-mail: wgong@lsec.cc.ac.cn.

<sup>‡</sup>Department of Mathematics, East China Normal University, Shanghai 200241, China. E-mail: lijiajie20120233@163.com

<sup>§</sup>Department of Mathematics & Shanghai Key Laboratory of Pure Mathematics and Mathematical Practice, East China Normal University, Shanghai 200241, China. E-mail: sfzhu@math.ecnu.edu.cn

partly because of its higher order discrete accuracy over the boundary formula, as discovered in [24] for elliptic problems, [47] for eigenvalue problems, and [46] for Stokes flows. The accuracy of approximate shape gradients could be essential for the implementation of numerical optimization algorithms [15] and may influence the convergence of shape optimization algorithms [6]. In [24], for the first time, the authors proved that the finite element approximations of shape gradients in boundary formulations of Eulerian derivatives have lower convergence rate with respect to the mesh size compared to the distributed one. It is further observed numerically in [29, 45] that the approximate shape gradient computed with the traditional boundary formulation may fail to converge to the optimal target domain, possibly because of the poor accuracy of the approximate shape gradients that may do not ensure the descent of the gradient. To alleviate the accuracy loss for the approximations of boundary type shape gradients, we proposed in [17] a modified boundary shape gradient for elliptic Dirichlet problems based on the discrete variational normal derivatives of the state and adjoint equations. In [17], a priori error estimate for the finite element approximations of the boundary type shape gradients with the proposed boundary corrections is derived which shows higher order convergence rate and accuracy than those of the classical one, and is comparable to the distributed one when the solutions are sufficiently smooth. However, the effectiveness of this improved boundary formula in shape optimization problems is not justified. In this paper, we test the performance of the boundary corrected shape gradient in two types of shape optimization algorithms: the scalar  $H^1$  shape gradient flow method (Algorithm 1) and the Fourier boundary parametrization method (Algorithm 2), which is the first contribution of this work. Numerical experiments show that this boundary correction can increase the robustness and effectiveness of shape gradient algorithms for shape optimizations, and can be more efficient and insensitive to the initial guess compared to the classical one. By presenting the corresponding discrete variational normal derivatives, we also extend this kind of corrections to shape optimizations of the Laplace eigenvalues ([4, 7, 32]), the Stokes eigenvalues [14, 34], the elliptic interface identification problem [2, 35], and the Stokes or Navier-Stokes flows ([30, 46]). These generalizations of the improved boundary type shape gradient to several typical and benchmark shape optimization problems form the second contribution of this work. Nevertheless, a disadvantage of the boundary formula and improved boundary formula is that it seems to be not straightforward to compute the Eulerian derivatives in the framework of automated differentiation as in [16, 36].

We remark that another advantage of our proposed boundary shape gradient lies in its continuity along the boundary. In [38, 40] the authors pointed out that due to the discontinuity of the discrete shape gradient, the  $L^2$ -projection has to be performed to bring back the continuity. However, the purpose of the projection is only to obtain a continuous shape gradient but the numerical accuracy is not improved. The abstract framework to obtain a descent direction based on the shape gradient is to solve an additional equation in an appropriate inner product (cf. [11] for different approaches). We have mentioned above that one advantage of the boundary type shape gradient over the distributed one is that it naturally offers an explicit choice of descent direction. However, the discrete boundary shape gradient in its classical form cannot be directly used in certain shape optimization algorithms because of its discontinuity, say, e.g., the Hadamard type, the Stefan-like and the surface diffusion flows, all of which use certain extensions of the shape gradient [11]. For the level set method or the computationally more efficient narrow band version [12, 42], one can either use a gradient flow based on the distributed Eulerian derivative [28] or use the extension of the boundary type shape gradient, where the continuity is indispensable for the finite-difference upwind schemes used in the discretization of the Hamilton-Jacobi equation [11, 42]. For the level set method combined with the proposed improved boundary shape gradient, the level set function can cut elements in the finite element mesh. One can assemble the boundary mass matrices in the discrete variational formulations (2.13)-(2.14) by using the Dirac delta of level set functions, so that the integrals can be performed on a strip domain near the boundary. For the ease of computing the gradient flow, the improved boundary Eulerian derivative can also be extended to a strip domain near the boundary or even the whole domain by virtue of the level set function. From the above concerns our proposed boundary type shape gradients for Dirichlet problems will be favorable for such shape optimization algorithms. Moreover, the third contribution of this work is to use the gradient recovery technique to bring back the continuity of the boundary type shape gradient for Neumann problems, which leads to an improvement over the classical boundary formula. As proved in [17], the classical discrete boundary type shape gradient for Neumann problems shares the similar convergence rate with that of the distributed one, but is discontinuous along the boundary. Our proposed boundary type shape gradient brings back the continuity and retains the convergence rate. We remark that the implementation of the gradient recovery techniques presented in Section 2.2 for Neumann boundary value problems is simple yet has been already included in commercial softwares.

The remaining of this paper is organized as follows. In §2 we recall our main ideas of improved boundary type shape gradient for Dirichlet boundary value problems, and propose for Neumann boundary value problems a modified boundary shape gradient that preserves continuity by using the gradient recovery technique. In §3, we extend our ideas of improved boundary type shape gradient for elliptic Dirichlet problems to different kinds of PDE-constrained shape optimization problems, including the eigenvalue optimization, the optimal shape design of fluid flows, the interface identification, etc. In §4 we test the numerical accuracy of the proposed boundary type shape gradients for both Dirichlet and Neumann boundary value problems. In §5 extensive numerical examples are presented to show the effectiveness of the improved shape gradient in two popular shape optimization algorithms.

A brief conclusion follows.

Let  $\Omega$  be an open bounded domain in  $\mathbb{R}^d$  ( $d = 2, 3$ ) with Lipschitz boundary  $\partial\Omega$ . Let  $W^{m,p}(\Omega)$  and  $W^{m,p}(\partial\Omega)$  be the usual Sobolev spaces defined on the domain  $\Omega$  and its boundary  $\partial\Omega$ , which are denoted respectively by  $H^m(\Omega)$  and  $H^m(\partial\Omega)$  when  $p = 2$ . Set  $H_0^1(\Omega) := \{v \in H^1(\Omega) : v = 0 \text{ on } \partial\Omega\}$ . For  $\Sigma = \Gamma$  or  $\partial\Omega$ , we denote by  $\langle \cdot, \cdot \rangle_\Sigma$  the inner product in  $L^2(\Sigma)$  associated with the norm  $\|\cdot\|_{L^2(\Sigma)}$ , or the duality pairing between  $H^{-\frac{1}{2}}(\Sigma)$  and  $H^{\frac{1}{2}}(\Sigma)$ . We use bold font to denote vector-valued spaces or variables. Set  $\mathbf{H}^m(\Omega) := H^m(\Omega)^d$  and  $\mathbf{H}_0^1(\Omega) := \{\mathbf{v} \in \mathbf{H}^1(\Omega) : \mathbf{v} = 0 \text{ on } \partial\Omega\}$ . Let  $(\cdot, \cdot)$  denote the inner product in  $L^2(\Omega)$  or  $\mathbf{L}^2(\Omega) := L^2(\Omega)^d$ . Denote by  $\|\cdot\|$  the Euclidean norm of a vector or the Frobenius norm of a matrix.

**2. Shape optimization for elliptic boundary value problems.** In this section we take the shape optimization for elliptic boundary value problems as an example to illustrate our main ideas. We propose modified boundary type shape gradients for both cases of Dirichlet and Neumann boundary conditions. We distinguish in the following the Dirichlet and Neumann boundary conditions for boundary correction and gradient recovery, respectively.

**2.1. Boundary correction for elliptic Dirichlet boundary value problems.** We consider the following PDE-constrained *shape functional*  $J(\cdot) : \Omega \mapsto \mathbb{R}$

$$J(\Omega) = \int_{\Omega} j(\mathbf{x}, u(\mathbf{x})) d\mathbf{x}, \quad (2.1)$$

where  $j : \Omega \times \mathbb{R} \rightarrow \mathbb{R}$  is  $C^1$  with its partial derivative  $j'_u(\mathbf{x}, u) := \frac{\partial}{\partial u} j(\mathbf{x}, u)$  being locally Lipschitz continuous,  $u : \Omega \rightarrow \mathbb{R}$  is the solution of the elliptic state equation with Dirichlet boundary condition

$$\begin{cases} -\Delta u + u = f & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega, \end{cases} \quad (2.2)$$

in which the functions  $f$  and  $g$  are assumed to belong to  $L^2(\mathbb{R}^d)$  and  $H^2(\mathbb{R}^d)$ , respectively. Here,  $f$  and  $g$  are identified with their restrictions onto  $\Omega$  and  $\partial\Omega$ , respectively. For compliance minimization problems we choose  $j = fu$ ; while for shape inverse problems we set simply  $j = \frac{1}{2}(u - u_{\text{obs}})^2$  for some observation data  $u_{\text{obs}}$ .

An essential ingredient of shape optimizations is the shape sensitivity analysis which measures the variations of the shape functional with respect to shape variations of the domain. We consider the *velocity method* for shape calculus (more details can be found in, e.g., [15, 41]). Let  $D \subset \mathbb{R}^d$  ( $\Omega \subset D$ ) be a hold-all domain, whose boundary  $\partial D$  is piecewise  $C^1$ . Denote

$$V^1(D) = \{\mathbf{v} \in \mathcal{D}^1(\mathbb{R}^d; \mathbb{R}^d) \mid \mathbf{v} \cdot \mathbf{n} = 0 \text{ a.e. on } \partial D \text{ except for the singular points } \bar{\mathbf{x}} \text{ of } \partial D, \\ \mathbf{v}(\bar{\mathbf{x}}) = 0 \text{ for all singular points } \bar{\mathbf{x}}\},$$

where  $\mathcal{D}^1(\mathbb{R}^d; \mathbb{R}^d)$  denotes the space of differentiable functions, whose derivatives vanish at infinity. For a time variable  $t \in [0, \tau]$  with  $\tau > 0$ , we choose a velocity field  $\mathbf{v} = \mathbf{v}(t, \mathbf{x}) \in C([0, \tau]; V^1(D))$ . For any  $\mathbf{X} \in \bar{D}$  and  $t \in [0, \tau]$ , let  $\mathbf{x} = \mathbf{x}(t, \mathbf{X})$  satisfy the following flow system

$$\frac{d\mathbf{x}}{dt}(t, \mathbf{X}) = \mathbf{v}(t, \mathbf{x}(t, \mathbf{X})), \quad \mathbf{x}(0, \mathbf{X}) = \mathbf{X}. \quad (2.3)$$

Let  $T_t(\mathbf{v}) : \bar{D} \rightarrow \bar{D}$  with  $T_t(\mathbf{v})(\mathbf{X}) := \mathbf{x}(t, \mathbf{X})$  be the associated one-to-one transformation from  $\bar{D}$  onto  $\bar{D}$ . Denote  $\Omega_t = T_t(\mathbf{v})(\Omega)$ . Now we define the *Eulerian derivative* of  $J(\Omega)$  at  $\Omega$  in the direction  $\mathbf{v}$  as follows

$$dJ(\Omega; \mathbf{v}) := \lim_{t \searrow 0} \frac{J(\Omega_t) - J(\Omega)}{t}. \quad (2.4)$$

For a given vector field  $\mathbf{v} = \mathbf{v}(t, \mathbf{x})$  in  $C([0, \tau]; \mathcal{D}^1(\mathbb{R}^d, \mathbb{R}^d))$ , the Eulerian derivative  $dJ(\Omega; \mathbf{v})$  of  $J$  in the direction  $\mathbf{v}$  can be formulated as an integral over a volume for the so-called *distributed shape gradient*, as well as an integral on the boundary for the so-called *boundary type shape gradient*. Explicit formula for the Eulerian derivative of  $J(\Omega)$  relies on the solution  $p$  of the following adjoint problem

$$\begin{cases} -\Delta p + p = j'_u(\cdot, u) & \text{in } \Omega, \\ p = 0 & \text{on } \partial\Omega. \end{cases} \quad (2.5)$$

Then we have the volumetric type Eulerian derivative (cf. [24, 20]):

$$dJ(\Omega; \mathbf{v}) = \int_{\Omega} \left( \nabla u \cdot (D\mathbf{v} + D\mathbf{v}^T) \nabla p - f \mathbf{v} \cdot \nabla p + \nabla_{\mathbf{x}} j(\cdot, u) \cdot \mathbf{v} + \text{div} \mathbf{v} (j(\cdot, u) - \nabla u \cdot \nabla p - up) + (j'_u(\cdot, u) - p) (\nabla g \cdot \mathbf{v}) - \nabla p \cdot \nabla (\nabla g \cdot \mathbf{v}) \right) d\mathbf{x}. \quad (2.6)$$

Moreover, if the boundary is smooth enough, the Eulerian derivative can be equivalently written as the boundary formula (cf. [24]):

$$dJ^D(\Omega; \mathbf{V}) = \int_{\partial\Omega} \mathbf{V}_{\mathbf{n}} \left( j(\cdot, u) + \partial_{\mathbf{n}} p \partial_{\mathbf{n}}(u - g) \right) ds, \quad (2.7)$$

where  $\mathbf{V}_{\mathbf{n}} := \mathbf{V} \cdot \mathbf{n}$ .

The distributed integral (2.6) and the boundary integral (2.7) are equivalent representations of the Eulerian derivative when the boundary is smooth. However, the formula (2.7) is more popular in the literature due to the Hadamard-Zolésio structure theorem, which says that the boundary type formula admits a representative  $\mathcal{G}(\Omega)$  in the space of distributions  $\mathcal{D}^k(\partial\Omega)$ , i.e., if  $\partial\Omega$  is smooth, there holds

$$dJ^D(\Omega; \mathbf{V}) = \langle \mathcal{G}(\Omega), \gamma_{\partial\Omega} \mathbf{V} \cdot \mathbf{n} \rangle_{\mathcal{D}^k(\partial\Omega)}, \quad (2.8)$$

where  $\gamma_{\partial\Omega} \mathbf{V} \cdot \mathbf{n}$  is the trace for the normal component of  $\mathbf{V}$  on the boundary  $\partial\Omega$ . This implies that only normal displacements of the boundary have an impact on the Eulerian derivative. Moreover, setting  $\mathbf{V} = -\mathcal{G}(\Omega)\mathbf{n}$  in (2.8) simply gives an explicit descent direction.

However, the above two formulae (2.6) and (2.7) have different numerical behaviors on the discrete level with, e.g., finite-element method. Let  $\{\mathcal{T}_h\}_{h>0}$  be a family of quasi-uniform and shape regular simplex triangulations of domain  $\Omega$ . For the domain with curved boundary we rely on the parametric finite elements (cf. [9, Section 10.4]) so that the elements may contain curved boundaries. Associated with  $\mathcal{T}_h$  we construct the finite element space  $V_h$ , which consists of continuous piecewise polynomials of first-order such that  $V_h \subset H^1(\Omega)$ . Denote  $V_h^0 := V_h \cap H_0^1(\Omega)$  and by  $V_h(\partial\Omega)$  the restriction of  $V_h$  to the boundary  $\partial\Omega$ .

The discretized boundary type Eulerian derivative for the shape optimization problem reads:

$$d_h J^D(\Omega; \mathbf{V}) = \int_{\partial\Omega} \mathbf{V}_{\mathbf{n}} \left( j(\cdot, u_h) + \partial_{\mathbf{n}} p_h (\partial_{\mathbf{n}} u_h - \partial_{\mathbf{n}} g) \right) ds, \quad (2.9)$$

where  $u_h \in V_h$  and  $p_h \in V_h^0$  solve the discrete state and adjoint state equations, respectively:

$$u_h \in V_h, \quad u_h|_{\partial\Omega} = g_h : \quad (\nabla u_h, \nabla v_h) + (u_h, v_h) = (f, v_h) \quad \forall v_h \in V_h^0, \quad (2.10)$$

$$p_h \in V_h^0 : \quad (\nabla v_h, \nabla p_h) + (p_h, v_h) = (j'_u(\cdot, u_h), v_h) \quad \forall v_h \in V_h^0 \quad (2.11)$$

with  $g_h$  being an approximation of  $g$  on  $\partial\Omega$ . The formula (2.9) is extensively used in the literature and let us call it *the classical boundary formula*. In [24] the authors compared the numerical performance of two discretized shape gradients, one with a volumetric formulation while the other with a boundary formulation, and showed that the volume type discrete shape gradient has better approximation property for state equations with Dirichlet boundary conditions compared to the counterpart of boundary integrals.

In [17] a modified discrete shape gradient was proposed, which will be referred to as the improved formula

$$\tilde{d}_h J^D(\Omega; \mathbf{V}) = \int_{\partial\Omega} \mathbf{V}_{\mathbf{n}} \left( j(\cdot, u_h) + \partial_{\mathbf{n}}^h p_h (\partial_{\mathbf{n}}^h u_h - \partial_{\mathbf{n}} g) \right) ds, \quad (2.12)$$

where  $\partial_{\mathbf{n}}^h u_h \in V_h(\partial\Omega)$  and  $\partial_{\mathbf{n}}^h p_h \in V_h(\partial\Omega)$  solve the following discrete variational problems

$$\langle \partial_{\mathbf{n}}^h u_h, v_h \rangle_{\partial\Omega} = (\nabla u_h, \nabla v_h) + (u_h, v_h) - (f, v_h) \quad \forall v_h \in V_h, \quad (2.13)$$

$$\langle \partial_{\mathbf{n}}^h p_h, v_h \rangle_{\partial\Omega} = (\nabla p_h, \nabla v_h) + (p_h, v_h) - (j'_u(\cdot, u_h), v_h) \quad \forall v_h \in V_h, \quad (2.14)$$

respectively. We have shown that the above modified discrete shape gradients have comparable approximation property to that of the volume integrals, i.e.,

$$|dJ^D(\Omega; \mathbf{V}) - \tilde{d}_h J^D(\Omega; \mathbf{V})| \lesssim h^2 \|\mathbf{V}_{\mathbf{n}}\|_{H^{3/2}(\partial\Omega)}, \quad (2.15)$$

if we use piecewise linear finite element approximations to the state and adjoint state variables.

The modified boundary formulae (2.13)-(2.14) are motivated by the variational normal derivatives for elliptic Dirichlet boundary value problems. In fact, for the state equation (2.2) it holds that  $\partial_{\mathbf{n}} u \in H^{-\frac{1}{2}}(\partial\Omega)$  such that

$$\langle \partial_{\mathbf{n}} u, v \rangle_{\partial\Omega} = (\nabla u, \nabla v) + (u, v) - (f, v) \quad \forall v \in H^1(\Omega). \quad (2.16)$$

Similarly, we have the variational problem for the outward normal derivative of solution to the adjoint equation (2.5): Find  $\partial_{\mathbf{n}} p \in H^{-\frac{1}{2}}(\partial\Omega)$  such that

$$\langle \partial_{\mathbf{n}} p, v \rangle_{\partial\Omega} = (\nabla p, \nabla v) + (p, v) - (j'_u(\cdot, u), v) \quad \forall v \in H^1(\Omega). \quad (2.17)$$

Therefore, on the discrete level we can define the discrete variational normal derivatives  $\partial_{\mathbf{n}}^h u_h$  and  $\partial_{\mathbf{n}}^h p_h$  which are better approximations to  $\partial_{\mathbf{n}} u$  and  $\partial_{\mathbf{n}} p$  compared to  $\partial_{\mathbf{n}} u_h$  and  $\partial_{\mathbf{n}} p_h$ . Moreover,  $\partial_{\mathbf{n}}^h u_h$  and  $\partial_{\mathbf{n}}^h p_h$  appearing in

(2.12) are solved in the space  $V_h(\partial\Omega)$ , the trace of a continuous finite element space. Thus, they are piecewise linear and continuous along the boundary  $\partial\Omega$ . This is in contrast to the piecewise constant and discontinuous approximations  $\partial_{\mathbf{n}}u_h$  and  $\partial_{\mathbf{n}}p_h$ . On the other hand, since  $j(\cdot, \cdot)$  in (2.12) is continuous with respect to the two arguments and  $u_h$  is continuous, so  $j(\cdot, u_h)$  is continuous. Therefore, the integrand  $(j(\cdot, u_h) + \partial_{\mathbf{n}}^h p_h(\partial_{\mathbf{n}}^h u_h - \partial_{\mathbf{n}}g))$  is continuous.

Furthermore, as pointed out in [17], the additional work in the modified discrete shape gradient is very little. In fact, only the linear systems with small numbers of Dofs are required to be solved, while the evaluation of the right-hand side in (2.13) and (2.14) is only performed in the neighboring elements of the boundary nodes because of the discrete state (2.10) and adjoint (2.11).

**2.2. Gradient recovery for elliptic Neumann boundary value problems.** In this subsection we consider the elliptic Neumann boundary value problem

$$\begin{cases} -\Delta u + u = f & \text{in } \Omega, \\ \partial_{\mathbf{n}}u = g & \text{on } \partial\Omega, \end{cases} \quad (2.18)$$

where  $f \in H^1(\mathbb{R}^d)$  and  $g \in H^2(\mathbb{R}^d)$ . Here,  $f$  and  $g$  are identified with their restrictions onto  $\Omega$  and  $\partial\Omega$ , respectively. The variational formulation of (2.18) is to find  $u \in H^1(\Omega)$  such that

$$(\nabla u, \nabla v) + (u, v) = (f, v) + \langle g, v \rangle_{\partial\Omega} \quad \forall v \in H^1(\Omega). \quad (2.19)$$

On the other hand, the variational formulation for the adjoint state equation reads: Find  $p \in H^1(\Omega)$  such that

$$(\nabla p, \nabla v) + (p, v) = (j'_u(\cdot, u), v) \quad \forall v \in H^1(\Omega). \quad (2.20)$$

Then,  $p$  satisfies in the distribution sense that

$$\begin{cases} -\Delta p + p = j'_u(\cdot, u) & \text{in } \Omega, \\ \partial_{\mathbf{n}}p = 0 & \text{on } \partial\Omega. \end{cases} \quad (2.21)$$

If  $\Omega$  is  $C^2$ , with the help of the above adjoint equation we can calculate the Eulerian derivative for the Neumann problem (cf. [24]):

$$dJ^N(\Omega; \boldsymbol{\nu}) = \int_{\partial\Omega} \boldsymbol{\nu}_{\mathbf{n}} \left( j(\cdot, u) - \nabla u \cdot \nabla p - up + fp + \partial_{\mathbf{n}}(gp) + Kgp \right) ds, \quad (2.22)$$

where  $K := \operatorname{div} \mathbf{n}$  is the mean curvature.

Now we are in the position to formulate the finite element approximations to the state and adjoint equations. For the state equation it is to find  $u_h \in V_h$  such that

$$(\nabla u_h, \nabla v_h) + (u_h, v_h) = (f, v_h) + \langle g, v_h \rangle_{\partial\Omega} \quad \forall v_h \in V_h. \quad (2.23)$$

For the discretization of the adjoint state equation it is to find  $p_h \in V_h$  such that

$$(\nabla p_h, \nabla v_h) + (p_h, v_h) = (j'_u(\cdot, u_h), v_h) \quad \forall v_h \in V_h. \quad (2.24)$$

Then we can formulate the discrete shape gradient for shape optimization of the Neumann boundary value problem:

$$d_h J^N(\Omega; \boldsymbol{\nu}) = \int_{\partial\Omega} \boldsymbol{\nu}_{\mathbf{n}} \left( j(\cdot, u_h) - \nabla u_h \cdot \nabla p_h - u_h p_h + f p_h + \partial_{\mathbf{n}}(g p_h) + K g p_h \right) ds. \quad (2.25)$$

One can prove that the above boundary formula (2.25) admits nearly second-order convergence rate (in fact,  $O(h^2 |\log h|)$ ) for finite-element discretization in case that the solutions have sufficient regularity (cf. [17, Theorem 3.13]). However, the integrand in the above formula (2.25) is not globally continuous along the boundary  $\partial\Omega$ , which may result in difficulty in certain shape optimization algorithms. For example, the direct extension of shape gradient (Hadamard flow and Stefan-like flow) (cf. [11]) and the narrow-band level set method ([12]) require the continuity of the shape gradient which is obviously not satisfied by (2.25).

In the current paper, we propose to enhance the above boundary formula by a gradient recovery technique. It has been proved that under certain mesh conditions the recovered gradient can admit higher order accuracy than that of the classical gradient (cf. [44, 48]). Although we can not expect that the modified formula achieves higher order finite-element convergence in general cases, it is globally continuous along the boundary. More importantly, and will be seen later, the computation of the modified formula is very cheap and the corresponding module of gradient recovery is already included in many commercial software of numerical simulations.

Now we illustrate the main idea of gradient recovery by using the state equation (2.18) as an example (cf. [44, 48] for more details). Let  $u_h \in V_h$  be the discrete solution of the state equation. We are aiming to construct

a vector  $\mathbf{G}_h(u_h) \in V_h^d := \prod_{i=1}^d V_h$  with  $\mathbf{G}_h : V_h \rightarrow V_h^d$  the recovery operator. For each mesh grid point  $z$ , let  $\Phi_z$  be the vectorial nodal basis function and define

$$\mathbf{G}_h(u_h) := \sum_{z \in \mathcal{T}_h} \mathbf{G}_h(u_h)(z) \Phi_z. \quad (2.26)$$

The essential part for the above reconstruction is to obtain the nodal values  $\mathbf{G}_h(u_h)(z)$  for each grid point  $z$ . There are many ways for the construction of the recovery operator  $\mathbf{G}_h$ , including the local averaging technique, the local  $L^2$ -projection, the least-squares methods, etc. Taking the local averaging method as an example, we can define  $\mathbf{G}_h(u_h)(z)$  as

$$\mathbf{G}_h(u_h)(z) := \frac{1}{m_z} \sum_{z \in K, K \in \omega_z} \nabla u_h(z)|_K, \quad (2.27)$$

where  $m_z$  is the number of elements sharing  $z$  as a vertex,  $\omega_z$  denotes the element patch consisting of elements with  $z$  as a vertex. One can also use the weighted local averaging technique to extend the above ideas. For the local least-squares method, Zienkiewicz-Zhu type recovery (cf. [48]) is the most popular one, which sets  $\mathbf{G}_h(u_h)(z) = \sigma_z(z)$  such that

$$E(\sigma_z) = \min_{\mathbf{w} \in V_z^d} E(\mathbf{w}) := \sum_{K \in \omega_z} \left\| \int_K \mathbf{w} d\mathbf{x} - \int_K \nabla u_h d\mathbf{x} \right\|^2, \quad (2.28)$$

where  $V_z^d$  denotes the local vectorial finite element space in  $\omega_z$ . When  $z \in \partial\Omega$ , a few extra neighboring elements should be added to  $\omega_z$  such that  $\omega_z$  contains more than three elements. For the regular mesh and suitable choice of  $\omega_z$  one can show that  $\mathbf{G}_h(u_h)$  exists for any  $\nabla u_h \in \mathbf{L}^2(\Omega)$ . For our purpose, we do not need to reconstruct  $\mathbf{G}_h(u_h)(z)$  for each vertex, but only the boundary vertex because we are only interested in the values on the boundary  $\partial\Omega$ .

A similar gradient recovery is performed for the adjoint equation to obtain  $\mathbf{G}_h(p_h) \in V_h^d$ . Now the modified shape gradient reads:

$$\tilde{d}_h J^N(\Omega; \boldsymbol{\nu}) = \int_{\partial\Omega} \boldsymbol{\nu}_n \left( j(\cdot, u_h) - \mathbf{G}_h(u_h) \cdot \mathbf{G}_h(p_h) - u_h p_h + f p_h + \partial_n g p_h + g \mathbf{G}_h(p_h) \cdot \mathbf{n} + K g p_h \right) ds. \quad (2.29)$$

It is obvious that the integrand in the above formula (2.29) is continuous along the boundary if  $\partial\Omega$  is  $C^2$  for  $g|_{\partial\Omega} \neq 0$  ( $C^1$  for  $g|_{\partial\Omega} \equiv 0$ ) and parametric finite elements are used for the discretization of the state and adjoint state equations. In fact, similar to (2.12),  $j(\cdot, u_h)$  in (2.29) is continuous and the finite element solutions  $u_h$  and  $p_h$  in a continuous finite element space are also continuous. On the other hand,  $\mathbf{G}_h(u_h) \in V_h^d$  and  $\mathbf{G}_h(p_h) \in V_h^d$  are continuous from the construction. In conclusion, the integrand in (2.29) is continuous.

If we do not use the parametric finite element method, then the integrand of the discrete shape gradient formula defined over the approximate boundary  $\partial\Omega_h$  has to be composed with the mapping operator between  $\partial\Omega$  and  $\partial\Omega_h$  to ensure the continuity, we refer to [17, Eq. (3.10) and (3.60)] for a similar definition. We remark that the above gradient recovery technique can also be used for distributed shape gradient if we are looking for a continuous discrete shape gradient.

**REMARK 2.1.** *We remark that the above improved discrete boundary type shape gradient formulae are defined in the form of parametric finite elements, as done in [24]. The counterparts without parametric finite elements for curved boundary were defined in [17, Eq. (3.10) and (3.60)] based on boundary polygonal approximations, where the a priori error estimates were also given. If we use non-parametric finite elements without curved boundary, say, polygonal domains, we can not expect the similar convergence rate as in the curved boundary case, although we still have the continuity of the outward normal derivatives under certain conditions (cf. [17, Remark 3.11]). We refer to [17, Remark 3.11] for more detailed discussions on the related convergence behaviors. However, numerical results in [17, Example 1 and Example 3] showed better convergence rates than what we expected.*

**3. Applications to shape optimizations with Dirichlet boundary conditions.** In this section we consider the applications of the improved boundary formulae for Dirichlet problems presented in Section 2 to different shape optimization problems, including the eigenvalue optimization, the interface identification, and the shape design in fluid flows.

**3.1. Laplace eigenvalue optimization.** We consider the Laplace eigenvalue problem with homogeneous Dirichlet boundary condition:

$$\begin{cases} -\Delta u = \lambda u & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \quad (3.1)$$

The variational formulation of problem (3.1) is to find  $\lambda \in \mathbb{R}^+$ ,  $0 \neq u \in H_0^1(\Omega)$  such that

$$(\nabla u, \nabla v) = \lambda(u, v) \quad \forall v \in H_0^1(\Omega). \quad (3.2)$$

It is well-known that there exists a sequence of eigenpairs  $(\lambda_\ell, u_\ell)$  with

$$0 < \lambda_1 \leq \lambda_2 \leq \dots \lambda_\ell \leq \dots < \infty,$$

where each eigenvalue is repeated according to its multiplicity and eigenfunctions are normalized by  $(u_i, u_j) = \delta_{ij}$  ( $i, j \geq 1$ ) with  $\delta_{ij}$  being the Kronecker delta. The eigenvalue  $\lambda_\ell$  can be characterized by the Rayleigh quotient, i.e., the Courant-Fischer formula

$$\lambda_\ell = \min_{\substack{S_\ell \subset H_0^1(\Omega) \\ \text{subspace of dim } \ell}} \max_{0 \neq v \in S_\ell} \frac{\int_\Omega |\nabla v|^2 d\mathbf{x}}{\int_\Omega v^2 d\mathbf{x}}.$$

The finite-dimensional weak form of (3.1) is to find  $\lambda_h \in \mathbb{R}^+$ ,  $0 \neq u_h \in V_h^0$  such that

$$(\nabla u_h, \nabla v_h) = \lambda_h(u_h, v_h) \quad \forall v_h \in V_h^0. \quad (3.3)$$

It is well-known that there exists a sequence of eigenpairs  $(\lambda_{\ell,h}, u_{\ell,h})$  for (3.3) with

$$0 < \lambda_{1,h} \leq \lambda_{2,h} \leq \dots \lambda_{\ell,h} \leq \dots < \infty,$$

where each eigenvalue is repeated according to its multiplicity and the eigenfunctions are normalized by  $(u_{i,h}, u_{j,h}) = \delta_{ij}$  ( $i, j \geq 1$ ).

We consider the following eigenvalue shape optimization problem subject to a volume constraint:

$$\min_{\text{Vol}(\Omega)=C} \lambda_\ell(\Omega), \quad (3.4)$$

where  $\text{Vol}(\Omega)$  denotes the volume of  $\Omega$  and  $C > 0$  is a prescribed number. Problem (3.4) is well understood for  $\ell = 1$  or 2 (cf. [23]). Many theoretical problems are still open for higher eigenvalues (see, e.g., [4, 32] for the numerical simulations). Eigenvalue optimization has fundamental applications in, e.g., the structural mechanics [5]. By homothety [23], which means that the eigenvalues of the Laplacian operator are invariant to rigid transformations of the domain, the original model (3.4) is equivalent to the unconstrained optimization problem [4, 32]

$$\min_{\Omega} \text{Vol}(\Omega)^{\frac{2}{d}} \lambda_\ell(\Omega). \quad (3.5)$$

The boundary formulation of the Eulerian derivative of problem (3.5) with  $J^D(\Omega) = \text{Vol}(\Omega)^{\frac{2}{d}} \lambda_\ell(\Omega)$  writes (cf. [23, 47])

$$dJ^D(\Omega; \boldsymbol{\nu}) = \int_{\partial\Omega} \left[ \frac{2}{d} \lambda_\ell \text{Vol}(\Omega)^{\frac{2}{d}-1} - \text{Vol}(\Omega)^{\frac{2}{d}} (\partial_{\mathbf{n}} u_\ell)^2 \right] \boldsymbol{\nu}_{\mathbf{n}} ds \quad (3.6)$$

with the corresponding discrete version (cf. [47])

$$d_h J^D(\Omega; \boldsymbol{\nu}) = \int_{\partial\Omega} \left[ \frac{2}{d} \lambda_{\ell,h} \text{Vol}(\Omega)^{\frac{2}{d}-1} - \text{Vol}(\Omega)^{\frac{2}{d}} (\partial_{\mathbf{n}}^h u_{\ell,h})^2 \right] \boldsymbol{\nu}_{\mathbf{n}} ds. \quad (3.7)$$

Now we introduce the improved boundary type shape gradients. As an improved approximation to  $\partial_{\mathbf{n}} u$  compared to  $\partial_{\mathbf{n}} u_h$ , the discrete variational outward normal derivative solves the following equation on the  $\ell$ -th eigenpair: Find  $\partial_{\mathbf{n}}^h u_{\ell,h} \in V_h(\partial\Omega)$  such that

$$\langle \partial_{\mathbf{n}}^h u_{\ell,h}, v_h \rangle_{\partial\Omega} = (\nabla u_{\ell,h}, \nabla v_h) - \lambda_{\ell,h}(u_{\ell,h}, v_h) \quad \forall v_h \in V_h, \quad (3.8)$$

which is a discrete counterpart of the continuous one:

$$\langle \partial_{\mathbf{n}} u_\ell, v \rangle_{\partial\Omega} = (\nabla u_\ell, \nabla v) - \lambda_\ell(u_\ell, v) \quad \forall v \in H^1(\Omega). \quad (3.9)$$

Now the improved boundary type shape gradient is given by

$$\tilde{d}_h J^D(\Omega; \boldsymbol{\nu}) = \int_{\partial\Omega} \left[ \frac{2}{d} \lambda_{\ell,h} \text{Vol}(\Omega)^{\frac{2}{d}-1} - \text{Vol}(\Omega)^{\frac{2}{d}} (\partial_{\mathbf{n}}^h u_{\ell,h})^2 \right] \boldsymbol{\nu}_{\mathbf{n}} ds. \quad (3.10)$$

**3.2. Stokes eigenvalue optimization.** Let  $\Omega \subseteq \mathbb{R}^d$  be a bounded open domain with boundary of class  $C^2$ . We consider the following Stokes eigenvalue problem:

$$\begin{cases} -\Delta \mathbf{u} + \nabla w = \lambda \mathbf{u} & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega, \\ \mathbf{u} = 0 & \text{on } \partial\Omega, \end{cases} \quad (3.11)$$

where  $\mathbf{u} = (u_1, \dots, u_d)^T : \Omega \rightarrow \mathbb{R}^d$  denotes the velocity field and  $w : \Omega \rightarrow \mathbb{R}$  is the pressure. The weak formulation of (3.11) writes: Find  $(\lambda, \mathbf{u}, w) \in \mathbb{R}^+ \times \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega)$  such that for  $\forall (\mathbf{v}, q) \in \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega)$  there holds

$$\begin{cases} \int_{\Omega} \mathbf{D}\mathbf{u} : \mathbf{D}\mathbf{v} \, d\mathbf{x} - \int_{\Omega} \operatorname{div} \mathbf{v} w \, d\mathbf{x} = \lambda \int_{\Omega} \mathbf{u} \cdot \mathbf{v} \, d\mathbf{x}, \\ \int_{\Omega} \operatorname{div} \mathbf{u} q \, d\mathbf{x} = 0, \end{cases} \quad (3.12)$$

where  $L_0^2(\Omega) := \{v \in L^2(\Omega) \mid \int_{\Omega} v \, d\mathbf{x} = 0\}$ . It is well known that problem (3.12) admits a sequence of positive eigenvalues  $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{\ell} \leq \dots$  tending to infinity as  $\ell \rightarrow \infty$ . The eigen-velocities  $\{\mathbf{u}_{\ell}\}_{\ell \geq 1} \subset \mathbf{H}_0^1(\Omega)$  and the eigen-pressures  $\{w_{\ell}\}_{\ell \geq 1} \subset L^2(\Omega)$  may be taken so that  $\{\mathbf{u}_{\ell}\}_{\ell \geq 1}$  constitutes an orthonormal basis of  $\mathcal{H}(\Omega) := \{\mathbf{v} \in \mathbf{L}^2(\Omega) : \nabla \cdot \mathbf{v} = 0 \text{ in } \Omega, \mathbf{v} = 0 \text{ on } \partial\Omega\}$ .

We consider the Stokes eigenvalue optimization problem

$$\min_{\operatorname{Vol}(\Omega)=C} \lambda_{\ell}(\Omega), \quad (3.13)$$

where  $\ell = 1, 2, \dots$  [14, 34]. As shown in [27] the two-dimensional Stokes eigenvalue problem is equivalent to the eigenvalue problem of the clamped buckling plate [3]. To the best of authors' knowledge the numerical methods for (3.13) are not available in the literature. The volume-constraint can be dealt with the augmented Lagrangian method. Introduce a Lagrangian functional:

$$\mathcal{L}(\Omega, l, \beta) := \lambda_{\ell} + l(\operatorname{Vol}(\Omega) - C) + \frac{1}{2}\beta(\operatorname{Vol}(\Omega) - C)^2, \quad (3.14)$$

where  $l > 0$  denotes the Lagrange multiplier and  $\beta > 0$  is a penalty parameter. When  $\lambda_{\ell}$  is shape differentiable and simple, we have the following first order optimality condition

$$\begin{cases} \mathcal{L}'(\Omega, l, \beta; \mathbf{V}) := \lambda'(\Omega, \mathbf{V}) + (l + \beta(\operatorname{Vol}(\Omega) - C))\operatorname{Vol}(\Omega)'(\Omega; \mathbf{V}) = 0, \\ \operatorname{Vol}(\Omega) - C = 0, \end{cases} \quad (3.15)$$

where the subscript  $\ell$  is omitted for notational brevity and

$$\lambda'(\Omega, \mathbf{V}) = - \int_{\partial\Omega} \|\mathbf{D}\mathbf{u}\|^2 \mathbf{V}_{\mathbf{n}} \, ds$$

by [34, p. 1008] and [14, Lemma 2.8]. The boundary type Eulerian derivative is given by

$$dJ^D(\Omega; \mathbf{V}) = \mathcal{L}'(\Omega, l, \beta; \mathbf{V}) = \int_{\partial\Omega} [l + \beta(\operatorname{Vol}(\Omega) - C) - \|\mathbf{D}\mathbf{u}\|^2] \mathbf{V}_{\mathbf{n}} \, ds. \quad (3.16)$$

Let  $\mathbf{V}_h \times W_h \subset \mathbf{H}^1(\Omega) \times L^2(\Omega)$  be the stable pair of finite element space for the Stokes equation which can be chosen as the Taylor-Hood element or the MINI element, etc. Let  $\mathbf{V}_h(\partial\Omega)$  be the trace of  $\mathbf{V}_h$  on the boundary  $\partial\Omega$  and  $\mathbf{V}_h^0 := \mathbf{V}_h \cap \mathbf{H}_0^1(\Omega)$ . The discretization of (3.12) reads: Find  $(\lambda_h, \mathbf{u}_h, w_h) \in \mathbb{R}^+ \times \mathbf{V}_h^0 \times W_h$  such that

$$\begin{cases} \int_{\Omega} \mathbf{D}\mathbf{u}_h : \mathbf{D}\mathbf{v}_h \, d\mathbf{x} - \int_{\Omega} \operatorname{div} \mathbf{v}_h w_h \, d\mathbf{x} = \lambda_h \int_{\Omega} \mathbf{u}_h \cdot \mathbf{v}_h \, d\mathbf{x} \quad \forall \mathbf{v}_h \in \mathbf{V}_h^0, \\ \int_{\Omega} \operatorname{div} \mathbf{u}_h q_h \, d\mathbf{x} = 0 \quad \forall q_h \in W_h. \end{cases} \quad (3.17)$$

There exists a sequence of positive eigenvalues  $0 < \lambda_{1,h} \leq \lambda_{2,h} \leq \dots \leq \lambda_{\ell,h} \leq \dots$  and discrete eigenvelocities  $\{\mathbf{u}_{\ell,h}\}_{\ell \geq 1}$  and discrete eigenpressures  $\{w_{\ell,h}\}_{\ell \geq 1}$ . We omit  $\ell$  for brevity below.

The variational formulation for the stress force on the boundary reads: Find  $\mathbf{D}_{\mathbf{n}}^h \mathbf{u}_h \in \mathbf{V}_h(\partial\Omega)$  such that

$$\langle \mathbf{D}_{\mathbf{n}}^h \mathbf{u}_h, \mathbf{t}_h \rangle_{\partial\Omega} = (\mathbf{D}\mathbf{u}_h, \mathbf{D}\mathbf{t}_h) + (\nabla w_h, \mathbf{t}_h) - (\lambda_h \mathbf{u}_h, \mathbf{t}_h) \quad \forall \mathbf{t}_h \in \mathbf{V}_h. \quad (3.18)$$

Then we have the improved boundary type shape gradient

$$\tilde{d}_h J^D(\Omega; \mathbf{V}) = \int_{\partial\Omega} [l + \beta(\operatorname{Vol}(\Omega) - C) - \|\mathbf{D}_{\mathbf{n}}^h \mathbf{u}_h\|^2] \mathbf{V}_{\mathbf{n}} \, ds, \quad (3.19)$$

which is an improvement over the classical one

$$d_h J^D(\Omega; \mathbf{V}) = \int_{\partial\Omega} [l + \beta(\operatorname{Vol}(\Omega) - C) - \|\mathbf{D}\mathbf{u}_h\|^2] \mathbf{V}_{\mathbf{n}} \, ds. \quad (3.20)$$



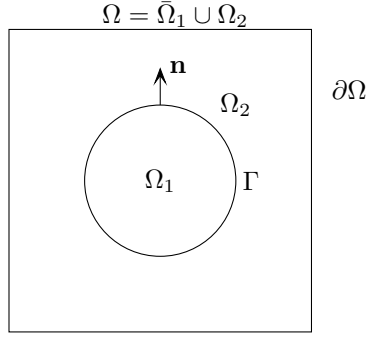


FIG. 3.1. An illustration of the interface problem.

**3.3. Interface identification.** In this subsection we consider the interface identification problem associated with the elliptic interface equation

$$\begin{cases} -\nabla \cdot (\sigma \nabla u) = f & \text{in } \Omega = \bar{\Omega}_1 \cup \Omega_2, \\ \llbracket u \rrbracket = 0 & \text{on } \Gamma, \\ \left[ \left[ \sigma \frac{\partial u}{\partial \mathbf{n}} \right] \right] = 0 & \text{on } \Gamma, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \quad (3.21)$$

where  $\sigma = \sigma(x)$  is a piecewise constant function

$$\sigma = \begin{cases} \sigma_1 & \text{in } \Omega_1, \\ \sigma_2 & \text{in } \Omega_2, \end{cases}$$

with  $\sigma_1$  and  $\sigma_2$  ( $\sigma_1 \neq \sigma_2$ ) being given positive constants, and  $\llbracket \cdot \rrbracket$  denotes the jump of a quantity across the interface  $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$  of subdomains  $\Omega_1$  and  $\Omega_2$ . Moreover, we assume that  $\Omega_1 \subset\subset \Omega$  and  $\Gamma = \partial\Omega_1$  (see Figure 3.1 for an illustration). Consider the objective functional

$$J(\Omega) = \int_{\Omega} j(u) d\mathbf{x}. \quad (3.22)$$

Our aim is to identify the interface  $\Gamma$ . Then we have the Eulerian derivative (cf. [35])

$$dJ^D(\Omega; \mathbf{V}) = \int_{\Gamma} \llbracket 2\sigma \partial_{\mathbf{n}} u \partial_{\mathbf{n}} p - \sigma \nabla u \cdot \nabla p \rrbracket \mathbf{V}_{\mathbf{n}} ds, \quad (3.23)$$

which can also be written as (cf. [2, Theorem 4.9])

$$dJ^D(\Omega; \mathbf{V}) = - \int_{\Gamma} (\sigma_1 - \sigma_2) \nabla_{\Gamma} u \cdot \nabla_{\Gamma} p \mathbf{V}_{\mathbf{n}} ds + \left( \frac{1}{\sigma_1} - \frac{1}{\sigma_2} \right) \int_{\Gamma} (\sigma_1 \partial_{\mathbf{n}} u^1) (\sigma_1 \partial_{\mathbf{n}} p^1) \mathbf{V}_{\mathbf{n}} ds \quad (3.24)$$

because of the homogeneous Neumann jump condition, where  $u^1 := u|_{\Omega_1}$  and  $p^1 := p|_{\Omega_1}$ , and  $p$  satisfies the adjoint equation

$$\begin{cases} -\nabla \cdot (\sigma \nabla p) = j'(u) & \text{in } \Omega = \bar{\Omega}_1 \cup \Omega_2, \\ \llbracket p \rrbracket = 0 & \text{on } \Gamma, \\ \left[ \left[ \sigma \frac{\partial p}{\partial \mathbf{n}} \right] \right] = 0 & \text{on } \Gamma, \\ p = 0 & \text{on } \partial\Omega. \end{cases} \quad (3.25)$$

We refer to [39, 40] for a parabolic interface problem which our ideas can also be applied.

On the discrete level, we need to discretize the state and adjoint equations. Let  $V_h^0 \subset H_0^1(\Omega)$  be a piecewise linear and continuous finite element space defined as above. We solve  $u_h, p_h \in V_h^0$  such that

$$(\sigma \nabla u_h, \nabla v_h) = \int_{\Omega} f v_h d\mathbf{x} \quad \forall v_h \in V_h^0, \quad (3.26)$$

$$(\sigma \nabla p_h, \nabla v_h) = \int_{\Omega} j'(u_h) v_h d\mathbf{x} \quad \forall v_h \in V_h^0. \quad (3.27)$$

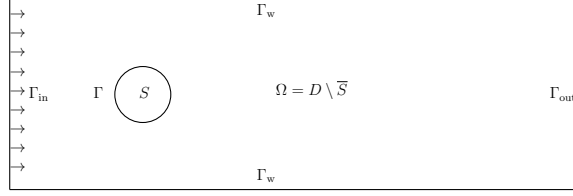


FIG. 3.2. An illustration of the geometric setting for problem (3.32) and Example 5.6.

Then we can compute the discrete variational normal derivatives  $\partial_{\mathbf{n}}^h u_h^1, \partial_{\mathbf{n}}^h p_h^1 \in V_h(\Gamma)$  such that

$$\langle \sigma_1 \partial_{\mathbf{n}}^h u_h^1, v_h \rangle_{\Gamma} = \int_{\Omega_1} \sigma_1 \nabla u_h^1 \nabla v_h \, d\mathbf{x} - \int_{\Omega_1} f v_h \, d\mathbf{x} \quad \forall v_h \in V_h(\Omega_1), \quad (3.28)$$

$$\langle \sigma_1 \partial_{\mathbf{n}}^h p_h^1, v_h \rangle_{\Gamma} = \int_{\Omega_1} \sigma_1 \nabla p_h^1 \nabla v_h \, d\mathbf{x} - \int_{\Omega_1} j'(u_h^1) v_h \, d\mathbf{x} \quad \forall v_h \in V_h(\Omega_1), \quad (3.29)$$

where  $V_h(\Gamma)$  and  $V_h(\Omega_1)$  denote the restriction of  $V_h$  to  $\Gamma$  and  $\Omega_1$ , respectively.

Now we can formulate the following discrete Eulerian derivative

$$\tilde{d}_h J^D(\Omega; \boldsymbol{\nu}) = - \int_{\Gamma} (\sigma_1 - \sigma_2) \nabla_{\Gamma} u_h \cdot \nabla_{\Gamma} p_h \boldsymbol{\nu}_{\mathbf{n}} \, ds + \left( \frac{1}{\sigma_1} - \frac{1}{\sigma_2} \right) \int_{\Gamma} (\sigma_1 \partial_{\mathbf{n}}^h u_h^1) (\sigma_1 \partial_{\mathbf{n}}^h p_h^1) \boldsymbol{\nu}_{\mathbf{n}} \, ds. \quad (3.30)$$

This is an improvement over the classical discrete formula

$$d_h J^D(\Omega; \boldsymbol{\nu}) = - \int_{\Gamma} (\sigma_1 - \sigma_2) \nabla_{\Gamma} u_h \cdot \nabla_{\Gamma} p_h \boldsymbol{\nu}_{\mathbf{n}} \, ds + \left( \frac{1}{\sigma_1} - \frac{1}{\sigma_2} \right) \int_{\Gamma} (\sigma_1 \partial_{\mathbf{n}} u_h^1) (\sigma_1 \partial_{\mathbf{n}} p_h^1) \boldsymbol{\nu}_{\mathbf{n}} \, ds, \quad (3.31)$$

where the classical formula is believed to be less accurate compared to the distributed one (cf. [35]).

**3.4. Shape optimization of Stokes flows.** In this subsection we consider the shape optimization of viscous incompressible fluid flows. Assume that the flow is stationary and the inertial force is negligible. Given  $\mathbf{f} \in \mathbf{H}^{-1}(\Omega)$ , the Stokes equation consists of seeking the velocity function  $\mathbf{u} \in \mathbf{H}_{*}^1(\Omega) := \{\mathbf{v} \in \mathbf{H}^1(\Omega) \mid \mathbf{v} = \mathbf{0} \text{ on } \Gamma \cup \Gamma_w, \mathbf{v} = \mathbf{u}_0 \text{ on } \Gamma_{\text{in}}\}$  and the pressure function  $w \in L_0^2(\Omega)$  such that

$$\begin{cases} -\mu \Delta \mathbf{u} + \nabla w = \mathbf{f} & \text{in } \Omega, \\ \operatorname{div} \mathbf{u} = 0 & \text{in } \Omega, \\ \mathbf{u} = \mathbf{0} & \text{on } \Gamma \cup \Gamma_w, \\ \mathbf{u} = \mathbf{u}_0 & \text{on } \Gamma_{\text{in}}, \\ (\mu D\mathbf{u} - pI)\mathbf{n} = \mathbf{0} & \text{on } \Gamma_{\text{out}}, \end{cases} \quad (3.32)$$

where  $\mu > 0$  denotes the viscosity coefficient and  $\partial\Omega = \Gamma \cup \Gamma_w \cup \Gamma_{\text{in}} \cup \Gamma_{\text{out}}$ . Consider the shape design for drag minimization (see Fig. 3.2) or energy dissipation of fluids (cf. [30])

$$\min_{\operatorname{Vol}(\Omega)=C} J(\Omega) = \mu \int_{\Omega} \|D\mathbf{u}\|^2 \, d\mathbf{x},$$

where the constant  $C > 0$  is prescribed. For the shape inverse problem (cf. [29, 43, 46]), we formulate it with a cost functional of output least-squares type

$$\min_{\Omega} J(\Omega) = \frac{1}{2} \int_{\Omega} \|\mathbf{u} - \mathbf{u}_d\|^2 \, d\mathbf{x},$$

where the observation  $\mathbf{u}_d$  is given.

The boundary formulations of the Eulerian derivatives for drag minimization and inverse problems are given (see, e.g., [30, 46] for the derivations)

$$dJ^D(\Omega; \boldsymbol{\nu}) = \int_{\Gamma} \left( l + \beta(\operatorname{Vol}(\Omega) - C) - \mu \|D\mathbf{u}\|^2 \right) \boldsymbol{\nu}_{\mathbf{n}} \, ds \quad (3.33)$$

and

$$dJ^D(\Omega; \boldsymbol{\nu}) = \int_{\Gamma} \left( \frac{1}{2} \|\mathbf{u} - \mathbf{u}_d\|^2 + \mu (D\mathbf{u}\mathbf{n}) \cdot (D\mathbf{v}\mathbf{n}) \right) \boldsymbol{\nu}_{\mathbf{n}} \, ds, \quad (3.34)$$

respectively, where  $l > 0$  denotes the Lagrange multiplier for the volume constraint and  $\beta > 0$  is the penalization parameter for the augmented Lagrangian formulation (cf. (3.14)), and  $(\mathbf{v}, r)$  is the solution of the adjoint problem

$$\begin{cases} -\mu\Delta\mathbf{v} + \nabla r = \mathbf{u} - \mathbf{u}_d & \text{in } \Omega, \\ \operatorname{div} \mathbf{v} = 0 & \text{in } \Omega, \\ \mathbf{v} = 0 & \text{on } \Gamma \cup \Gamma_w \cup \Gamma_{\text{in}}, \\ (\mu D\mathbf{v} - rI)\mathbf{n} = \mathbf{0} & \text{on } \Gamma_{\text{out}}. \end{cases} \quad (3.35)$$

Let  $\mathbf{V}_h \times W_h \subset \mathbf{H}^1(\Omega) \times L_0^2(\Omega)$  be the stable finite element spaces for solving Stokes equations defined in Section 3.2. We define  $\hat{\mathbf{V}}_h$  and  $\check{\mathbf{V}}_h$  as follows. For any  $\mathbf{v}_h \in \hat{\mathbf{V}}_h$  we assume that  $\mathbf{v}_h = 0$  on  $\Gamma \cup \Gamma_w \cup \Gamma_{\text{in}}$ , while for any  $\mathbf{u}_h \in \check{\mathbf{V}}_h$  let  $\mathbf{u}_h = 0$  on  $\Gamma \cup \Gamma_w$  and  $\mathbf{u}_h = \mathbf{u}_0^h$  on  $\Gamma_{\text{in}}$  with  $\mathbf{u}_0^h$  an approximation to  $\mathbf{u}_0$ . Let  $\mathbf{u}_h \in \check{\mathbf{V}}_h$  and  $\mathbf{v}_h \in \hat{\mathbf{V}}_h$  be the solutions to the state equation (3.32) and the adjoint (3.35), respectively. The corresponding discrete shape gradients of boundary type are given respectively by

$$d_h J^D(\Omega; \boldsymbol{\nu}) = \int_{\Gamma} \left( l + \beta(\operatorname{Vol}(\Omega) - C) - \mu \|\mathbf{D}\mathbf{u}_h \mathbf{n}\|^2 \right) \boldsymbol{\nu}_n ds \quad (3.36)$$

and

$$d_h J^D(\Omega; \boldsymbol{\nu}) = \int_{\Gamma} \left( \frac{1}{2} \|\mathbf{u}_h - \mathbf{u}_d\|^2 + \mu(\mathbf{D}\mathbf{u}_h \mathbf{n}) \cdot (\mathbf{D}\mathbf{v}_h \mathbf{n}) \right) \boldsymbol{\nu}_n ds. \quad (3.37)$$

Now we introduce the improved shape gradient of boundary type for the Stokes flow. Let  $\mathbf{V}_h(\Gamma)$  be the trace of  $\mathbf{V}_h$  on the boundary  $\Gamma$ . The variational formulation for the discrete variational outward normal derivatives reads: Find  $\mathbf{D}_n^h \mathbf{u}_h \in \mathbf{V}_h(\Gamma)$  such that

$$\mu \langle \mathbf{D}_n^h \mathbf{u}_h, \mathbf{t}_h \rangle_{\Gamma} = \mu(\mathbf{D}\mathbf{u}_h, \mathbf{D}\mathbf{t}_h) + (\nabla w_h, \mathbf{t}_h) - (\mathbf{f}, \mathbf{t}_h) \quad \forall \mathbf{t}_h \in \mathbf{V}_h. \quad (3.38)$$

For the shape inverse problem, the variational normal derivative for the adjoint equation reads: Find  $\mathbf{D}_n^h \mathbf{v}_h \in \mathbf{V}_h(\Gamma)$  such that

$$\mu \langle \mathbf{D}_n^h \mathbf{v}_h, \mathbf{t}_h \rangle_{\Gamma} = \mu(\nabla \mathbf{v}_h, \nabla \mathbf{t}_h) + (\nabla r_h, \mathbf{t}_h) - (\mathbf{u}_h - \mathbf{u}_d, \mathbf{t}_h) \quad \forall \mathbf{t}_h \in \mathbf{V}_h. \quad (3.39)$$

For the drag minimization problem, no adjoint problem is introduced. With the above preparations, the improved shape gradients read:

$$\tilde{d}_h J^D(\Omega; \boldsymbol{\nu}) = \int_{\Gamma} \left( l + \beta(\operatorname{Vol}(\Omega) - C) - \mu \|\mathbf{D}_n^h \mathbf{u}_h\|^2 \right) \boldsymbol{\nu}_n ds \quad (3.40)$$

for the drag-minimization problem and

$$\tilde{d}_h J^D(\Omega; \boldsymbol{\nu}) = \int_{\Gamma} \left( \frac{1}{2} \|\mathbf{u}_h - \mathbf{u}_d\|^2 + \mu(\mathbf{D}_n^h \mathbf{u}_h) \cdot (\mathbf{D}_n^h \mathbf{v}_h) \right) \boldsymbol{\nu}_n ds \quad (3.41)$$

for the shape inverse problem.

**3.5. Shape optimization of Navier-Stokes flows.** In this subsection we consider the shape optimization governed by the Navier-Stokes equations. We use a similar setting as in the above subsection, including the objective functionals for respectively the drag-minimization problem and the shape inverse problem. By shape sensitivity analysis, we have the following boundary Eulerian derivative for the drag/energy minimization problem [30]

$$dJ^D(\Omega; \boldsymbol{\nu}) = \mu \int_{\partial\Omega} ((\mathbf{D}\mathbf{u}\mathbf{n}) \cdot (\mathbf{D}\mathbf{v}\mathbf{n}) - \|\mathbf{D}\mathbf{u}\mathbf{n}\|^2) \boldsymbol{\nu}_n ds, \quad (3.42)$$

in which  $(\mathbf{u}, w)$  satisfies the Navier-Stokes equation

$$\begin{cases} -\mu\Delta\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla w = \mathbf{f} & \text{in } \Omega, \\ \operatorname{div} \mathbf{u} = 0 & \text{in } \Omega, \\ \mathbf{u} = 0 & \text{on } \partial\Omega \end{cases} \quad (3.43)$$

and  $(\mathbf{v}, r)$  satisfies the adjoint state

$$\begin{cases} -\mu\Delta\mathbf{v} - \mathbf{D}\mathbf{v}\mathbf{u} + \mathbf{D}\mathbf{u}^T\mathbf{v} + \nabla r = -2\mu\Delta\mathbf{u} & \text{in } \Omega, \\ \operatorname{div} \mathbf{v} = 0 & \text{in } \Omega, \\ \mathbf{v} = 0 & \text{on } \partial\Omega, \end{cases} \quad (3.44)$$

where homogeneous Dirichlet boundary conditions have been used for simplicity in the above formulations. Let  $\mathbf{u}_h \in \mathbf{V}_h^0$  and  $\mathbf{v}_h \in \mathbf{V}_h^0$  be the solutions to the state equation (3.43) and the adjoint problem (3.44), respectively, where  $\mathbf{V}_h^0$  is defined as in Section 3.2. The classical discrete shape gradient is given by

$$d_h J^D(\Omega; \mathcal{V}) = \mu \int_{\partial\Omega} ((D\mathbf{u}_h \mathbf{n}) \cdot (D\mathbf{v}_h \mathbf{n}) - \|D\mathbf{u}_h \mathbf{n}\|^2) \mathcal{V}_n ds. \quad (3.45)$$

Define the discrete variational outward normal derivative for the state equation: Find  $D_n^h \mathbf{u}_h \in \mathbf{V}_h(\partial\Omega)$  such that

$$\mu \langle D_n^h \mathbf{u}_h, \mathbf{t}_h \rangle_{\partial\Omega} = \mu(D\mathbf{u}_h, D\mathbf{t}_h) + (\mathbf{u}_h \cdot D\mathbf{u}_h, \mathbf{t}_h) + (\nabla w_h, \mathbf{t}_h) - (\mathbf{f}, \mathbf{t}_h) \quad \forall \mathbf{t}_h \in \mathbf{V}_h. \quad (3.46)$$

The discrete variational outward normal derivative for the adjoint equation reads: Find  $D_n^h \mathbf{v}_h \in \mathbf{V}_h(\partial\Omega)$  such that

$$\mu \langle D_n^h \mathbf{v}_h, \mathbf{t}_h \rangle_{\partial\Omega} = \mu(D\mathbf{v}_h, D\mathbf{t}_h) - (D\mathbf{v}_h \mathbf{u}_h, \mathbf{t}_h) + (D\mathbf{u}_h^T \mathbf{v}_h, \mathbf{t}_h) + (\nabla r_h, \mathbf{t}_h) - (2\mu D\mathbf{u}_h, D\mathbf{t}_h) \quad \forall \mathbf{t}_h \in \mathbf{V}_h. \quad (3.47)$$

With above preparations we are ready to give the improved shape gradient formula

$$\tilde{d}_h J^D(\Omega; \mathcal{V}) = \mu \int_{\partial\Omega} ((D_n^h \mathbf{u}_h) \cdot (D_n^h \mathbf{v}_h) - \|D_n^h \mathbf{u}_h\|^2) \mathcal{V}_n ds. \quad (3.48)$$

**REMARK 3.1.** *For the additional computational complexity of the modified boundary formula, we need to solve one (for self-adjoint optimization problems) or two additional boundary equations after obtaining the discrete state and possible adjoint state. This is not a big burden even when the state equation is nonlinear because, on the one hand, the boundary equation needs only be solved once when the state is solved with satisfactory accuracy by using nonlinear iterations, i.e., no nonlinear iterations are required for the boundary equations; on the other hand, the boundary equation has relatively smaller number of degrees of freedom compared to the state or adjoint state equations that are posed on the whole computational domain.*

**4. Accuracy test for the improved boundary type shape gradients.** In this section we present some numerical results to verify the convergence rates of finite element approximations to our modified boundary type shape gradients, including the accuracy and continuity improved formulae for Dirichlet problems and the continuity preserving formulae for Neumann problems.

**4.1. Accuracy enhancement for Dirichlet boundary value problems.** The shape gradient of the shape functional is a linear continuous operator on  $H^{3/2}(\mathbb{R}^d; \mathbb{R}^d)$  and belongs to its dual space in the boundary type Eulerian derivative. The continuous infinite-dimensional operator norm for the approximate shape gradients is difficult to compute numerically and we follow the ideas of [24] to approximate it by a tractable one on a finite-dimensional subspace. Let  $\gamma$  be an arbitrary positive integer as in [24]. We consider an approximate operator norm on a finite-dimensional space consisting of vector fields in  $\mathcal{P}_{\gamma, \gamma}(\mathbb{R}^d; \mathbb{R}^d) (\subset H^{3/2}(\mathbb{R}^d; \mathbb{R}^d))$ , whose components are multivariate polynomials of degree up to  $\gamma$ . Based on the equivalence of norms over finite-dimensional spaces, we replace the  $H^{3/2}$ -norm with a more tractable  $H^1$ -norm. Finally, we compute the approximate dual norm as

$$\mathcal{E} := \left( \max_{\mathbf{0} \neq \mathcal{V} \in \mathcal{P}_{\gamma, \gamma}(\mathbb{R}^d; \mathbb{R}^d)} \frac{|dJ^D(\Omega; \mathcal{V}) - \tilde{d}_h J^D(\Omega; \mathcal{V})|^2}{\|\mathcal{V}\|_{H^1(\Omega)}^2} \right)^{1/2}. \quad (4.1)$$

We take a global basis  $\{\mathcal{V}_i\}_{i=1}^\nu$  of vector fields in  $\mathcal{P}_{\gamma, \gamma}(\mathbb{R}^d; \mathbb{R}^d)$ , where  $\nu = dC_{\gamma+d}^d$  with  $C_{\gamma+d}^d$  denoting the combination coefficient. More precisely, we choose

$$\{\mathcal{V}_i\}_{i=1}^\nu = \{[\Pi_{i=1}^d x_i^{\beta_i}, 0, \dots, 0], \dots, [0, \dots, 0, \Pi_{i=1}^d x_i^{\beta_i}]\}_{\sum_{i=1}^d \beta_i \leq \gamma},$$

where  $\beta_i$  ( $i = 1, \dots, d$ ) is a non-negative integer (see [46, p. 141] for more details). Choose  $\gamma = 2$  and 3.

As the enhanced numerical accuracy of the improved boundary shape gradients has been verified in [17] for elliptic equations, we consider in this subsection the shape optimizations of Laplace eigenvalue problem and Stokes/Navier-Stokes flow.

**EXAMPLE 4.1.** *In the first example we consider the error accuracy test for the Laplace eigenvalue problem (3.2). We first set  $\Omega = (0, 1)^2$ , where the first eigen-pair  $(\lambda_1, u_1) = (2\pi^2, 2 \sin(\pi x_1) \sin(\pi x_2))$ . Then we choose  $\Omega$  to be an irregular wedge, on which the analytic eigenpair is not available and we take the approximated eigenpair on a very fine mesh as the reference.*

Both Fig. 4.1 and Fig. 4.2 show that the improved shape gradients have second-order convergence rate for the problem settings in Example 4.1, while the classical shape gradients have only first-order convergence rate for both regular and irregular domains. We expect that the convergence analysis in [17] can be adapted to the Laplacian eigenvalue problems.

**EXAMPLE 4.2.** *Consider the drag minimization problem for Stokes and Navier-Stokes flows as presented in Section 3.4 and 3.5. Set  $\Omega = (0, 1)^2$  and  $\mu = 1$ . For Stokes flow, the problem has an analytic solution  $\mathbf{u} = [u_1, u_2]^T$*

with  $u_1 = (\sin(\pi x_1))^2 \sin(\pi x_2) \cos(\pi x_2)$ ,  $u_2 = -(\sin(\pi x_2))^2 \sin(\pi x_1) \cos(\pi x_1)$  and  $p = \sin(2\pi x_1) \sin(2\pi x_2)$ . For the Navier-Stokes flow, the exact solutions are unknown. The approximate reference solutions are computed on a very fine mesh with the MINI finite element discretization. Newton's method is used for the linearization of the Navier-Stokes equation.

We can observe from Fig. 4.3 second-order convergence rate for the improved shape gradients, while the classical formula has only first-order convergence rate. The rigorous proof for this second-order convergence seems to be not straightforward compared to the elliptic problem in [17].

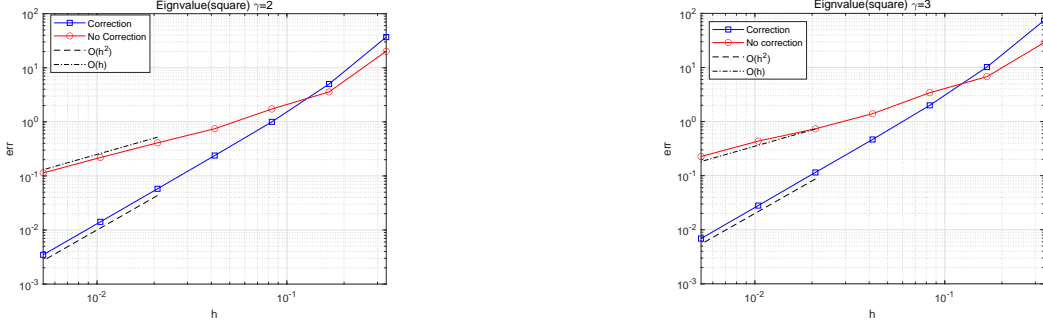


FIG. 4.1. Convergence history of the approximate shape gradient for the first Laplace eigenvalue optimization on a square in Example 4.1.

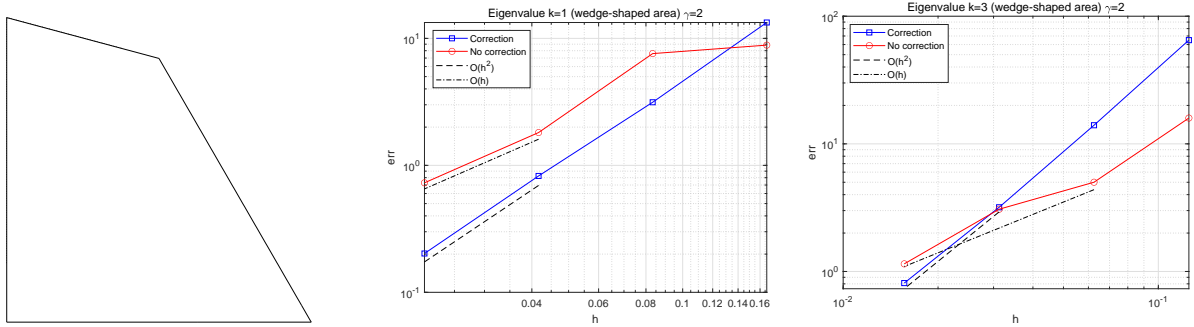


FIG. 4.2. Convergence history of the approximate shape gradient for Laplace eigenvalue optimization on a wedge-shaped domain (left) for Example 4.1 with  $\gamma = 2$ :  $\ell = 1$  (middle) and  $\ell = 3$  (right).

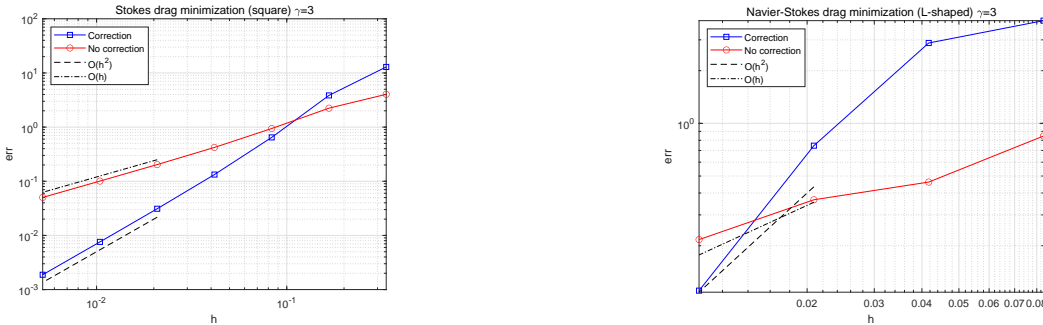


FIG. 4.3. Convergence history of the approximate shape gradients of drag minimization for Stokes flow (left) and Navier-Stokes flow (right) in Example 4.2.

**4.2. Continuity preserving for Neumann boundary value problems.** In this subsection we test the accuracy of the modified boundary shape gradient for Neumann boundary value problem (2.18). We consider the gradient recovery by using either local averaging (2.27) or superconvergent patch recovery (SPR) (2.28). We show that the modified formula has the same second-order convergence rate and slightly smaller errors compared to the classical formula. The illustration of the benefits of continuity preserving formulae over the discontinuous ones in shape optimization algorithms is delicate and rely on the choice of specific algorithms, this is not presented in this paper and will be exploited elsewhere. Consider  $J(\Omega) = \int_{\Omega} u^2 dx$ . In the following three examples, each domain is triangulated with a quasi-uniform/uniform mesh as illustrated in Fig. 4.4.

EXAMPLE 4.3. Set  $\Omega = (-1, 1)^2$ ,  $f = x_1^2 + x_2^2$  and  $g = 0$ .

EXAMPLE 4.4. Set a L-shaped domain  $\Omega = (-1, 1)^2 \setminus (-1, 0]^2$ . Set  $f = \sin(0.5\pi x_1) \sin(0.5\pi x_2)$  and  $g = 0$ .

EXAMPLE 4.5. Set  $\Omega = (-1, 1)^2$ . Choose  $f = (1 + 0.5\pi^2) \sin(0.5\pi x_1) \sin(0.5\pi x_2)$  and  $g = 0$  such that  $u = \sin(0.5\pi x_1) \sin(0.5\pi x_2)$ , while the adjoint state is given by  $p = 2 \sin(0.5\pi x_1) \sin(0.5\pi x_2) / (2 + \pi^2)$ .

Our numerical tests show very similar results for the approximate shape gradients with either local averaging method or SPR, so we only present parts of the numerical results. The left and right plots of Fig. 4.5 show that the approximate shape gradients with SPR have the same second-order convergence rate as the classical formula without gradient recovery for Example 4.3 and Example 4.5 on the square domain. For Example 4.4, the middle plot of Fig. 4.5 further shows that the approximate shape gradients with and without gradient recovery have nearly the same convergence rate of  $O(h^{1.5})$ , since the solution has limited regularity on the nonconvex L-shaped domain. Moreover, we observe for all three examples that the gradient-recovered shape gradients have slightly smaller errors.

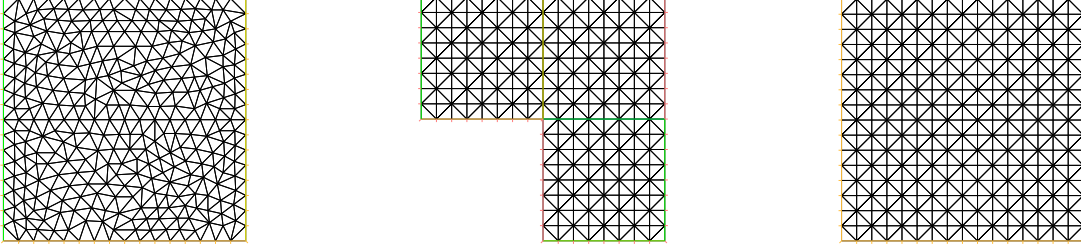


FIG. 4.4. Illustration of the computational domains and meshes for Examples 4.3-4.5 (from left to right).

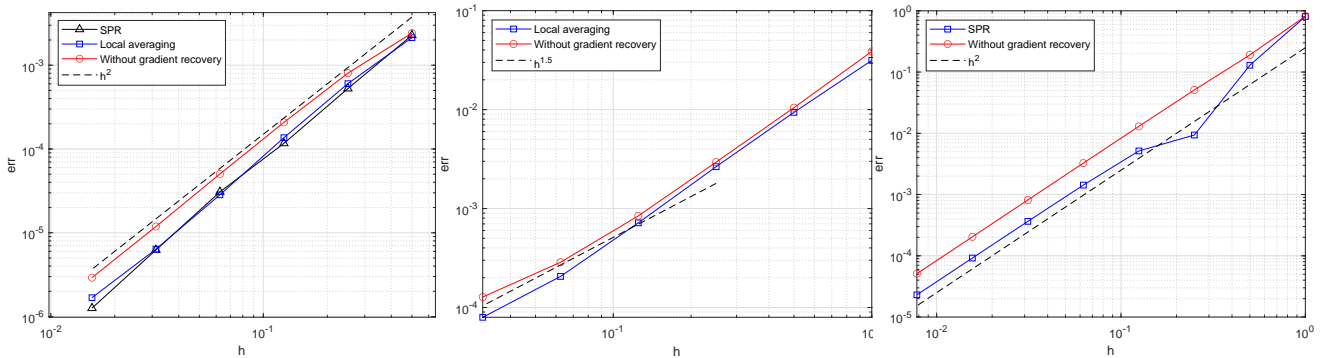


FIG. 4.5. Comparisons on the convergence history of the approximate shape gradient with local averaging and SPR for Examples 4.3-4.5 (from left to right).

**5. Numerical experiments for shape optimization problems.** To demonstrate the performance of the improved shape gradient for Dirichlet problems, we use a gradient descent algorithm to solve numerically the shape optimization models including the elliptic boundary value problem, the Laplace and Stokes eigenvalue optimization, the shape design in fluid flows and the interface identification. We apply improved shape gradients in both the traditional mesh moving algorithm (cf. [38, 45]) and the Fourier boundary parametrization method [7, 33, 43]. We propose two corresponding numerical algorithms below. Though both algorithms are based on the modified boundary shape gradient formula, they differ in the boundary representation and boundary variations for deformations. More specifically, a  $H^1$  shape gradient flow is used in Algorithm 1 to increase smoothness of the deformation field and realize the moving of the boundary by the movement of mesh grids of the domain. In Algorithm 2, the Fourier boundary parametrization method is used to represent the boundary and we are led to a finite dimensional optimization problem. The boundary variations and domain deformations are thus realized through the update of the Fourier coefficients. No gradient flow is required in Algorithm 2, however, a quasi-uniform mesh generation is required for each of the new domain.

**5.1. Mesh moving based shape gradient algorithm.** In the mesh moving algorithm we need to find the descent direction of mesh points by using the shape gradients. The smoothness of a descent direction can be enhanced by solving a shape gradient flow (see e.g., [28, 39, 45]). The  $H^1$ -type shape gradient flow associated with the boundary Eulerian derivative is to find  $\theta \in \mathbf{H}^1(\Omega)$  such that

$$\int_{\Omega} (D\theta : D\xi + \theta \cdot \xi) dx = -dJ^D(\Omega; \xi) \quad \forall \xi \in \mathbf{H}^1(\Omega), \quad (5.1)$$

where  $D\boldsymbol{\theta} : D\boldsymbol{\xi} = \sum_{i,j=1}^d \partial_j \theta_i \partial_j \xi_i$ . Let  $\mathbf{S}_h \subset \mathbf{H}^1(\Omega)$  be a linear and continuous finite element space. The finite-dimensional approximation of (5.1) with improved boundary shape gradient is to find  $\boldsymbol{\theta}_h \in \mathbf{S}_h$  such that

$$\int_{\Omega} (D\boldsymbol{\theta}_h : D\boldsymbol{\xi} + \boldsymbol{\theta}_h \cdot \boldsymbol{\xi}) d\mathbf{x} = -\tilde{d}_h J^D(\Omega; \boldsymbol{\xi}) \quad \forall \boldsymbol{\xi} \in \mathbf{S}_h. \quad (5.2)$$

The gradient flow system involves the vectorial field  $\boldsymbol{\theta}_h$  whose numerical computations are demanding, especially in three-dimensions. Here we introduce a *new* decomposition scheme which decomposes the  $d$ -dimensional vectorial problem (5.2) into  $d$  scalar problems to reduce the computational cost. More precisely, we can solve in parallel the variational problems: find  $\theta_{h,i} \in V_h$  such that

$$\int_{\Omega} (\nabla \theta_{h,i} \cdot \nabla \eta + \theta_{h,i} \eta) d\mathbf{x} = -\tilde{d}_h J^D(\Omega; \boldsymbol{\xi}_i) \quad \forall \eta \in V_h \quad (5.3)$$

for  $i = 1, 2, \dots, d$ , where  $\boldsymbol{\xi}_i = [0, \dots, \eta, \dots, 0]^T$  with all components being 0 except the  $i$ th being  $\eta$ . Set  $\boldsymbol{\theta}_h = [\theta_{h,1}, \theta_{h,2}, \dots, \theta_{h,d}]^T$ . In practice, we will not use parallel solvers considering that  $d$  is only 2 and 3. The  $d$  linear systems of scalar gradient flows after the Galerkin finite element discretization share the same stiffness and mass matrices and thus can be assembled and decomposed (Cholesky decomposition in 2D) only once. For  $d = 2$ , therefore, the computational costs of the second scalar linear system by a direct solver are much less than that of the first one. For  $d = 3$ , we can solve them by using a more efficient preconditioned conjugate gradient method.

A step size  $\delta > 0$  chosen at each iteration influences the efficiency of algorithms with or without boundary correction. Line-search strategies can be used to determine a suitable step size at each iteration. For simplicity when comparing the two algorithms, however, we choose a maximal constant step such that the objective decreases. Moreover, we find that the step size depending on the current mesh size should be chosen to prevent the transformed simplex from having negative area after deformation.

During the shape evolution, the domain size and the shapes of some simplex in the mesh may change significantly, which may deteriorate the mesh qualities such as the quasi-uniformness and shape-regularity. The poor quality of meshes may affect the accuracy of finite element approximations of the shape gradients. This probably leads to negative effects on the convergence of deformation algorithms. To overcome this problem, we adopt a re-meshing strategy when a new domain is formed by moving mesh grids. The computational cost of re-meshing can be ignored comparing with solving finite element equations. We present the shape gradient optimization method in the form of Algorithm 1. The algorithm is stopped by setting a fixed number of iteration or by the criterion

$$\frac{|J(\Omega_k) - J(\Omega_{k+1})|}{J(\Omega_k)} < \varepsilon,$$

where  $\Omega_k$  denotes the domain of  $k$ th iteration and  $\varepsilon > 0$  is a prescribed error tolerance. For eigenvalue optimization below, we denote by  $m$  the multiplicity of eigenvalue and adopt a strategy used in [4, 45] by considering the average Eulerian derivative of a multiple eigenvalue associated with all corresponding eigenfunctions.

---

**Algorithm 1:** Shape gradient descent method with improved boundary formula

---

```

Choose an initial domain and set  $\delta$ ;
while the iteration does not stop do
    Solve the state problem;
    Solve possible the adjoint problem;
    Solve the discrete outward normal derivatives of the state and adjoint;
    Solve the shape gradient flow;
    Deformation;
    while reversed triangle/tetrahedron occurs do
         $\delta \leftarrow \delta/2$ ;
        Deformation;
    end
    remeshing;
end

```

---

All numerical results below are obtained by using FreeFEM [25] on a laptop with Intel(R) Core(TM) i7-8750H CPU@2.20GHz and 24GB RAM.

**5.1.1. Elliptic boundary value problem.** In this subsection we consider the following **shape optimization problem** of elliptic equations:

$$\min_{\Omega} J(\Omega) = \frac{1}{2} \int_{\Omega} (u - u_d)^2 d\mathbf{x},$$

where  $u_d$  is a given observation data and  $u$  is the solution of (2.2) satisfying Dirichlet boundary condition.

EXAMPLE 5.1. Choose the target (optimal) domain as a unit disk:  $\{(x_1, x_2) | x_1^2 + x_2^2 < 1\}$ . We set  $u_d = 1 - x_1^2 - x_2^2$  such that  $f = 5 - x_1^2 - x_2^2$  and  $g = 0$ . Set  $\delta = 1$ . First, we choose a square  $(-0.8, 0.8)^2$  as the initial guess. In the first row of Fig. 5.1, we see that the shape gradient algorithm with improved boundary formula converges to the target optimal domain from the initial square. Second, we choose a L-shaped domain  $\Omega = (-1.5, 0.5)^2 \setminus \{[-0.5, 0.5] \times (-1.5, -0.5]\}$  as the initial guess. We can see from the second row of Fig. 5.1 that the algorithm with improved boundary formula converges to the target optimal domain again. For comparisons as shown in Fig. 5.2 (the first and second plots), the algorithm with classical boundary formula converges to the disk from the initial square, but fails to converge from the initial L-shaped domain even with re-meshing. Re-meshing ensures the improvement of mesh quality that is beneficial for deformations. The necessity of re-meshing step is demonstrated especially at corners, where the meshes deteriorate frequently during evolution. From Fig. 5.2 (the third and fourth plots) we observe that the algorithm with improved boundary formula but without re-meshing can converge to the disk from the initial square. However, it fails to converge from the initial L-shaped domain, which demonstrates the necessity of adopting re-meshing strategy. Fig. 5.3 shows that the shape gradient algorithm with improved boundary formula converges more efficiently and effectively than the algorithm with classical boundary formula.

EXAMPLE 5.2. Choose the target domain to be a square  $(-1, 1)^2$ . Set  $u_d = (1 - x_1^2)(1 - x_2^2)$  such that  $f = 4 - 2x_1^2 - 2x_2^2$ . Fig. 5.4 shows that both the initial disk and L-shaped domain converge to the target square by our algorithm. However, we find that the algorithm by using the classical boundary formula converges from the initial disk to the square, but fails to converge from the initial L-shaped domain. Fig. 5.5 shows that the classical and the present algorithms without re-meshing both converge to the optimal domain from the initial disk. However, the classical algorithm fails to converge to the target square from the initial L-shaped domain even with re-meshing. Fig. 5.6 shows the convergence history of the objective functional for this example with two types of initial domains.

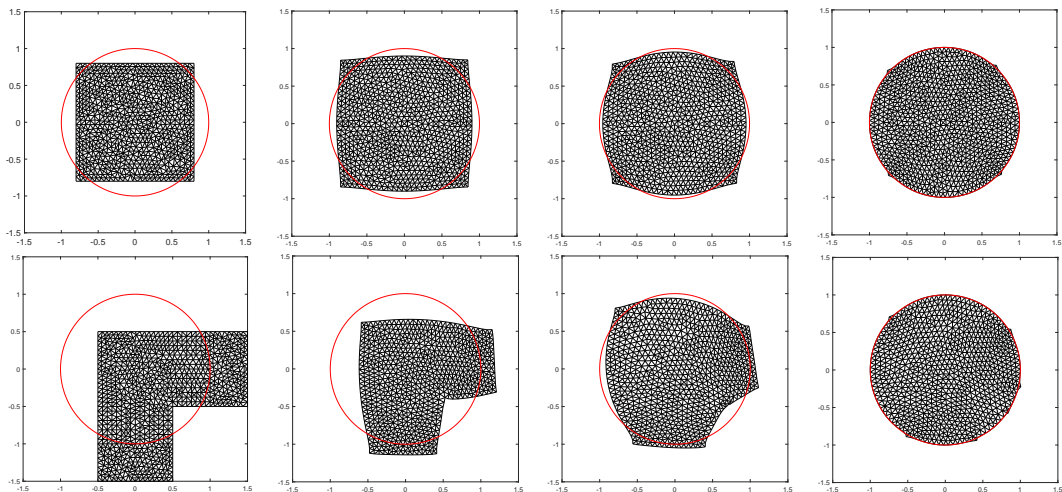


FIG. 5.1. Deformations from the initial domains (left) to the optimal disk (right) for Example 5.1 by Algorithm 1: initial square (first row) and initial L-shaped domain (second row).

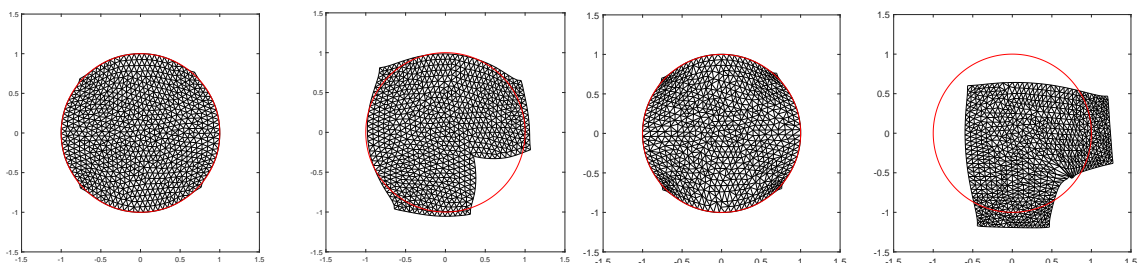


FIG. 5.2. Final shapes by shape gradient algorithm with the classical formula: initial square (first) and initial L-shaped domain (second); and final shapes by Algorithm 1 without re-meshing: initial square (third) and initial L-shaped domain (fourth) for Example 5.1.

We compare the numerical performance of our proposed decomposed shape gradient flow (5.3) of the (scalar) elliptic operator (“Elliptic+Decomposition” in Table 5.1) with that of the  $H^1$  deformation gradient flow associated with the linear elasticity operator (“Elasticity” in Table 5.1, see, e.g., [26, 40]). Choose the penalty parameter



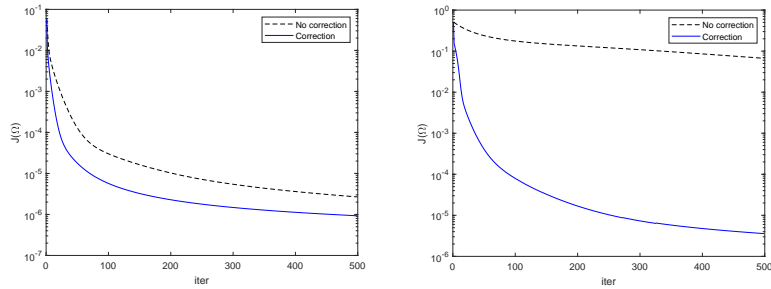


FIG. 5.3. Comparisons on the convergence history of the objective functional between algorithms with the improved and the classical boundary formulae: initial square (left) and initial L-shaped domain (right) for Example 5.1.

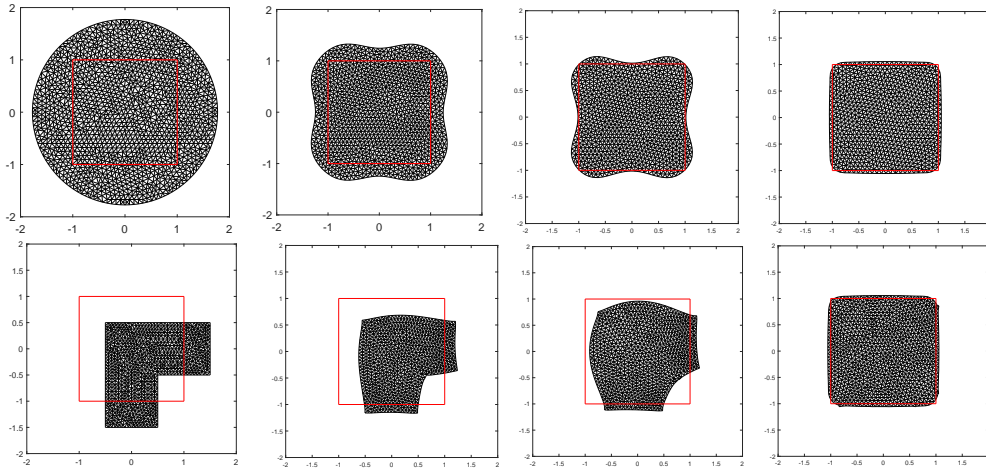


FIG. 5.4. Deformations from the initial domain (left) to the optimal square (right) for Example 5.2 by Algorithm 1: initial square (first row) and initial L-shaped domain (second row).

$\alpha$  in [26, Eq. (15)] to be 1. Set the Lamé parameters  $\mu$  and  $\lambda$  in [40, Eq. (5.2)] to be 1 and 0.5, respectively. Both gradient flows can increase the smoothness of the deformation field and make the objective value decrease. A key feature of the difference is that the elastic shape gradient flow cannot be decomposed into  $d$  separate scalar type elliptic problems. Besides, the gradient flow with the inner product based on conformal transformations and the symmetric part of the gradient (CT-H(sym)) [26] (“CT-H(sym)” in Table 5.1) also cannot be decomposed. Moreover, the CT technique [26] is limited to 2D.

In Fig. 5.7, we present the numerical results for Example 5.1 and Example 5.2 by using the same Algorithm 1 (500 iterations for the optimization loop) except that the gradient flow is replaced by four types of elastic shape gradient flows with different Eulerian derivatives and inner products: (1) boundary formula (“Boundary” in Table 5.1), (2) improved boundary formula (“Improved Boundary” in Table 5.1), (3) distributed shape gradient (“Volume” in Table 5.1), (4) distributed shape gradient and inner product based on CT-H(sym) [26]. The comparisons on final results show similar effectiveness of all the four gradient flows with the proposed gradient flow (5.3). For comparisons of computational costs among different gradient flows, Table 5.1 shows that although remeshing is required, the proposed decomposed  $H^1$  gradient flow with the improved boundary formula is the fastest due to the efficient solving of the scalar gradient flows. Under the same inner product (either  $H^1$  or elastic type), additional computational efforts are expected for the improved boundary formula compared with the standard boundary formula. For elastic gradient flows, the improved boundary formula is faster than the volume formulation. For boundary/improved boundary formulae, the  $H^1$  gradient flow even without scalar decomposition is more efficient than the elastic type.

**5.1.2. Laplacian eigenvalue optimization.** In this subsection we consider Laplacian eigenvalue optimization.

**EXAMPLE 5.3.** Consider the unconstrained volume formulation (3.5) for minimizing the first eigenvalue. Both 2D and 3D cases are presented to show the effectiveness of the improved boundary formula. For 2D, we choose a L-shaped domain  $\Omega = (-1.5, 0.5)^2 \setminus \{[-0.5, 0.5] \times (-1.5, -0.5]\}$  as the initial guess. For 3D, we choose a cube  $(-1, 1)^3$  removed by  $[-1, 0]^3$  as the initial domain. From Fig. 5.8 we see that the shape gradient algorithm with improved boundary formula converges to the target optimal domain from the initial L-shaped domain. Fig. 5.9 shows that the shape gradient algorithm with improved boundary formula converges efficiently. We refer to Table 5.2 and Fig. 5.10 for the computed results on the optimal objectives, the optimal shapes and convergence histories for different eigenvalues.

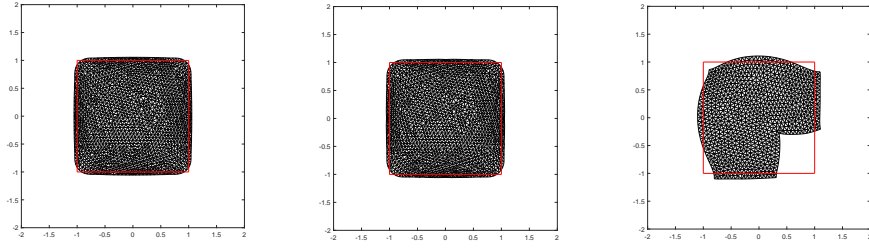


FIG. 5.5. Comparisons of final shapes by the shape gradient algorithms for Example 5.2: from the initial disk with improved formula (left) or with the classical formula (middle), both without re-meshing; from the L-shaped domain with re-meshing and the classical formula (right).

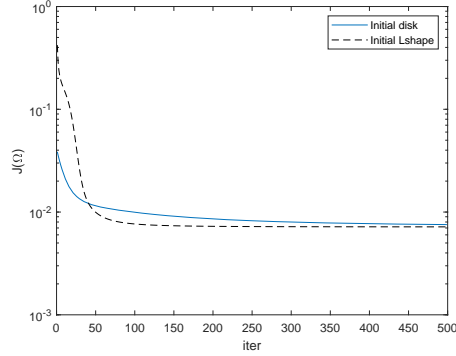


FIG. 5.6. Convergence history of the shape functional by Algorithm 1 for Example 5.2.

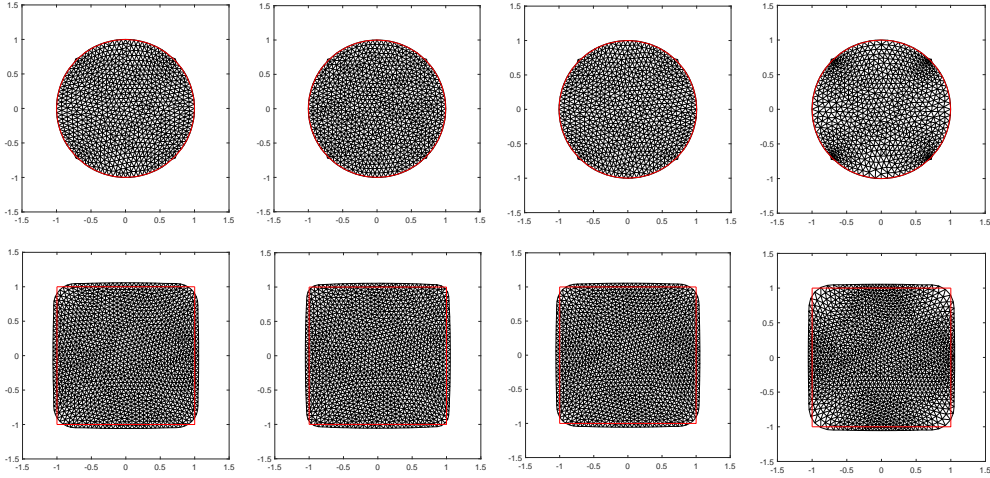


FIG. 5.7. Comparisons of final shapes from the initial square (first row) and initial disk (second row) by different shape gradient algorithms for Example 5.1 and Example 5.2, respectively: elastic gradient flow with classical boundary formula, elastic gradient flow with the improved boundary formula, elastic gradient flow with the distributed Eulerian derivative, and the CT-H(sym) flow with the distributed Eulerian derivative (from left to right).

Formula type	Example 5.1	Example 5.2
Elliptic+Boundary	32.15	56.01
Elliptic+Improved Boundary	44.06	78.11
Elliptic+Decomposition+Improved Boundary	28.21	69.95
CT-H(sym)+Improved Boundary	44.88	78.19
Elasticity+Boundary	44.36	70.51
Elasticity+Improved Boundary	51.62	88.6
Elasticity+Volume	53.86	103.4
CT-H(sym)+Volume	41.78	74.85

TABLE 5.1

Comparisons on computational costs (seconds) for Example 5.1 and Example 5.2 by Algorithm 1 among different shape gradient formulae: 778 and 1493 nodes for the initial square and initial disk, respectively.

We compare in Table 5.3 the numerical performance of the proposed improved boundary formula combined with the decomposed shape gradient flow (5.3) of scalar type with that of other types of shape gradient flows, all of which can increase the smoothness of the deformation field and make the objective value decrease. For fair comparisons, we fix the number of total iterations as the same, say 100. Choose the same initial square and initial cube for 2D and 3D, respectively. The computational times are collected by using the same Algorithm 1 except that the gradient flow is different. Table 5.3 shows for optimizing eigenvalues in 2D that the proposed shape gradient flow based on the improved boundary formula and decomposition is the fastest. The gradient flow (without decomposition) with improved boundary formula is faster than that with the volume Eulerian derivative (see [45] for numerical results of optimal domains, which coincide with those in Table 5.2). Assume the usage of the improved boundary formula, the more complicated elastic gradient flow is slower than the elliptic type gradient flow; while considering that no remeshing is needed, the gradient flow with CT-H(sym) as the inner product [26] is faster than the vectorial  $H^1$  gradient flow; lastly, the vectorial  $H^1$  gradient flow is faster than the elastic type. For 3D cases, Table 5.3 demonstrates that the improved boundary formula is faster than the volume type, although slower than the standard boundary type formula.

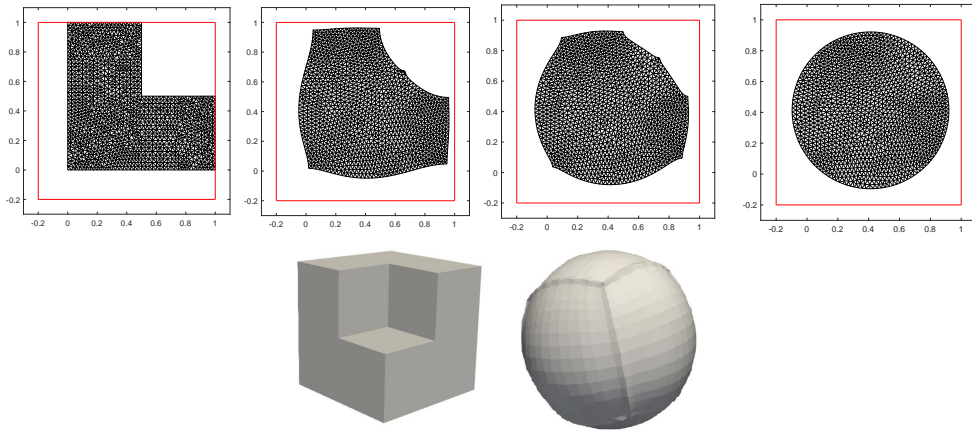


FIG. 5.8. Deformations from the L-shaped domain to the optimal disk and ball for Example 5.3 by Algorithm 1: 2D (first row) and 3D (second row).

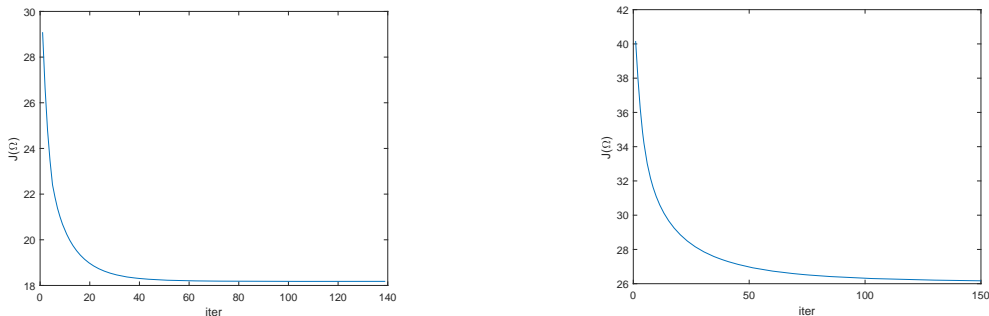


FIG. 5.9. Convergence history of  $\lambda_1$  by Algorithm 1 for Example 5.3: 2D (left) and 3D cases (right).

### 5.1.3. Shape design of fluid flows.

In this subsection we consider the optimal design of fluid flows. **EXAMPLE 5.4.** Consider the minimization of the energy dissipation in fluid flows with applications in, e.g., vascular design of blood flows (see Fig. 5.11). We consider both the Stokes and Navier-Stokes flows and choose a L-shaped domain (see Fig. 5.12) as the initial design. We choose a velocity  $(0.2 - 0.8x_2^2, 0)^T$  on the left boundary (inflow) and impose a traction-free condition on the right down boundary (outflow). Non-slip Dirichlet boundary conditions are imposed on other boundaries. Set  $\beta = 1$ ,  $\varepsilon = 0.001$  and  $C = 3$  in Algorithm 1. For both the Stokes and Navier-Stokes flows, we see from Fig. 5.12 that Algorithm 1 converges to the target optimal domain. For the Stokes case, Fig. 5.12 shows that the same algorithm except the gradient flow is replaced by the volumetric CT-H(sym) type [26] leads to quasi-uniform mesh-based domain deformations without remeshing. Fig. 5.13 shows that the shape gradient algorithm with improved boundary formula converges efficiently with a final objective slightly smaller than that obtained by using the classical boundary formula. The convergence history of the objective value by the distributed CT-H(sym) gradient flow [26] is plotted with the same number of total iterations being used as the proposed algorithm with improved boundary formulae. The volume errors reduce as the iteration numbers increase for both algorithms.

**EXAMPLE 5.5.** Consider the inverse problem for Stokes flow with homogeneous Dirichlet boundary conditions. Set the target domain as an annulus

$$\{(x_1, x_2) \mid x_1 = r \cos \theta, x_2 = r \sin \theta, \theta \in [0, 2\pi], r \in (r_1, r_2)\}.$$




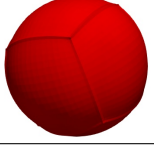

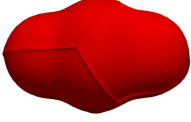

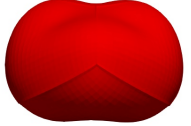

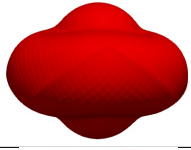

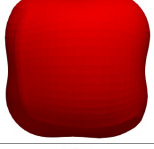

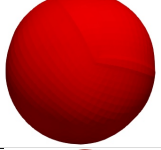

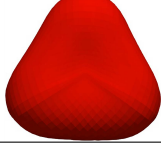
$\ell$	2D			3D		
	$m$	$\lambda_\ell$	Optimal shape	$m$	$\lambda_\ell$	Optimal shape
3	2	46.158		2	49.1844	
4	2	65.1573		3	52.5879	
5	2	78.744		3	65.2021	
6	3	89.8317		3	77.0325	
7	3	107.762		3	82.1911	
8	3	121.459		4	86.812	
9	3	134.284		5	89.5896	
10	4	143.622		6	93.0313	

TABLE 5.2  
Numerical results for Example 5.3 on the Laplace eigenvalue optimization.

Formula type	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	$\lambda_5$	$\lambda_6$	$\lambda_7$	$\lambda_8$	$\lambda_9$	$\lambda_{10}$
		2D: 4435		vertices	for	the	initial	square		
Elliptic+Volume	46.56	46.70	45.46	47.44	46.65	47.46	43.24	45.53	47.78	47.05
Elliptic+Boundary	36.82	36.24	36.32	38.63	36.93	39.58	37.80	39.12	38.48	37.06
Elliptic+Improved Boundary	40.07	38.46	39.42	40.50	39.36	39.74	40.97	43.67	41.35	39.74
Elliptic+Decomposition										
+Improved Boundary	36.13	33.62	34.76	33.85	33.13	33.58	31.98	35.69	35.78	35.66
C-T-H(sym)+Improved Boundary	37.38	37.29	36.65	37.45	37.00	37.27	38.06	38.31	38.05	37.88
Elasticity+Improved Boundary	42.18	41.03	42.88	42.52	43.65	43.73	42.59	43.04	42.62	43.56
		3D: 9261		vertices	for	the	initial	cube		
Elliptic+Volume	1141	1130	1149	1151	1146	1152	1129	1134	1137	1117
Elliptic+Boundary	834.6	828.2	836.7	827.3	836.6	839.7	826.1	823.6	839.2	841.3
Elliptic+Improved Boundary	871.3	875.5	866.4	861.3	874.7	866.7	869.6	868.2	885.2	882.6

TABLE 5.3  
Comparisons on computational costs (seconds) for Example 5.3 by Algorithm 1 among different shape gradient formulae.

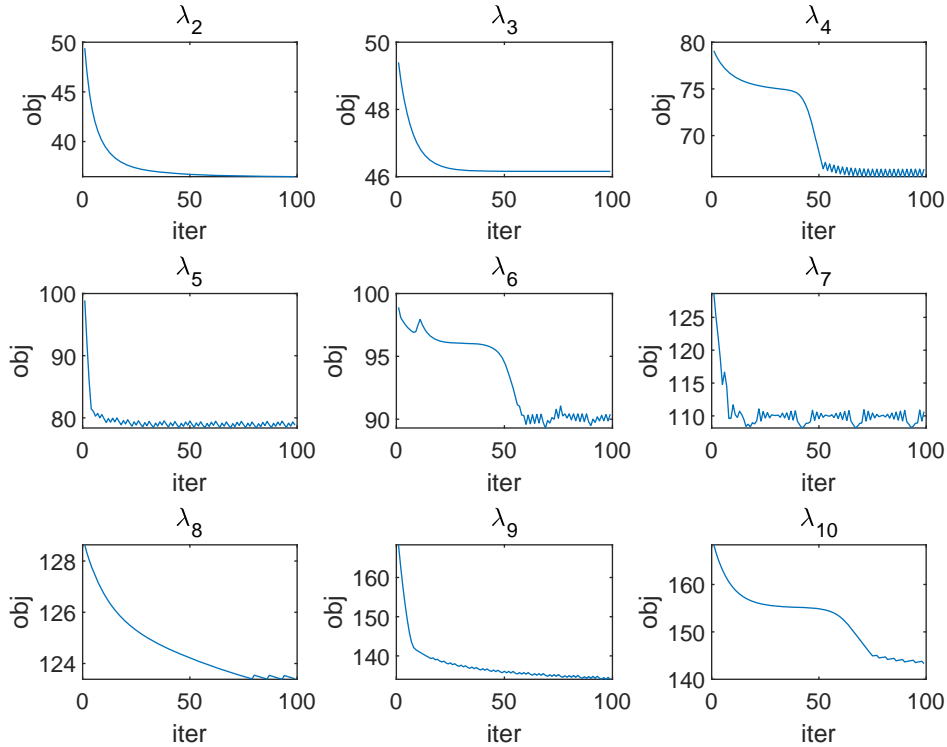


FIG. 5.10. Convergence histories of the Laplace eigenvalues by Algorithm 1 for Example 5.3 (2D and  $\ell = 2, 3, \dots, 10$ ).

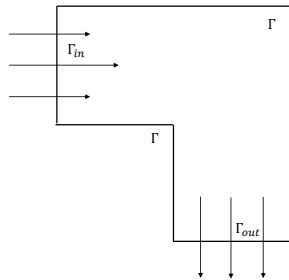


FIG. 5.11. Illustration of the geometric setting for the energy dissipation minimization of fluid flows in Example 5.4.

Choose  $\varepsilon = 10^{-5}$  in Algorithm 1 and the viscosity coefficient  $\mu = 0.01$ . Let

$$\mathbf{u}_d = [-x_2(r^2 - r_1^2)(r^2 - r_2^2), x_1(r^2 - r_1^2)(r^2 - r_2^2)]^T,$$

where  $r^2 = \|\mathbf{x}\|^2 = x_1^2 + x_2^2$ . Choose  $r_1 = 0.25$  and  $r_2 = 1$ . The inner boundary is the unknown part to be identified, while the outer boundary is fixed. We use the rectangle  $(-0.5, 0.5) \times (-0.4, 0.4)$  as the initial guess. In Fig. 5.14, we see that the shape gradient algorithm with improved boundary formula converges to the target domain from the square. Fig. 5.15 shows that the shape gradient algorithm with improved boundary formula converges efficiently. However, the shape gradient algorithm with classical boundary formula fails to converge to the optimal shape, possibly because the inaccurate calculation of the descent direction at the sharp corners caused by low accuracy of the shape gradient leads to errors for grid moving.

EXAMPLE 5.6. Consider the drag minimization of Stokes and Navier-Stokes flows on the exterior of a 2D obstacle as shown in Fig. 3.2. Boundary conditions are the same as in (3.32) with  $\mathbf{u}_0 = ((0.5 - x_2)(0.5 + x_2), 0)^T$ . Set  $\beta = 10$ ,  $\varepsilon = 10^{-3}$  and  $C = 1.95$ . Fig. 5.16 shows that the optimal shape of Stokes flow by Algorithm 1 agrees well with that obtained by the volumetric CT-H(sym) gradient flow [26]. See Fig. 5.17 for the optimal shape of the Navier-Stokes flow. We also refer to Fig. 5.18 for the convergence histories of the objective functional and the volume error.

Using the same total iteration number of the optimization loop, the comparisons made in Table 5.4 on computational costs of the Stokes case show that although the proposed algorithm using a *vectorial* gradient flow with improved boundary formulae is slower, the proposed algorithm with the *scalar* gradient flow is faster than the algorithm using the volumetric CT-H(sym) gradient flow [26]. Scalar decompositions can save significant computational efforts for solving the proposed gradient flow, while scalar decompositions cannot be done for the gradient flow of [26].

EXAMPLE 5.7. Consider the drag minimization on the exterior of a 3D obstacle with Algorithm 1. Let  $\mathbf{u}_0 = (1, 0, 0)^T$ . Choose  $C = 1.966$ . Set  $\beta = 5$ ,  $\beta = 2.5$  for the Stokes case and the Navier-Stokes case, respectively. In this example we fix the total iteration number. In Fig. 5.19 we present the initial domain and the optimal domains for both the Stokes and Navier-Stokes cases, while the convergence histories of the objective functional and the volume error can be found in Fig. 5.20.

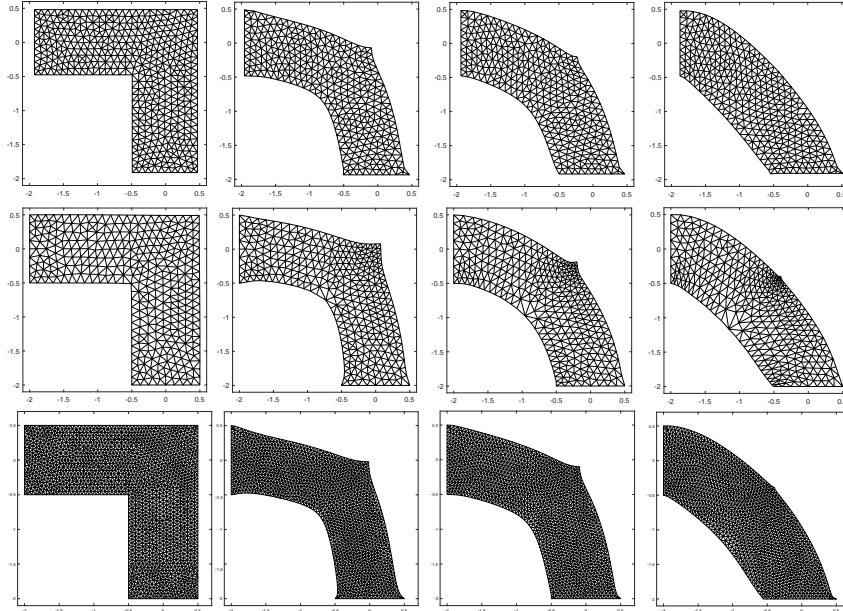


FIG. 5.12. Deformations from the initial L-shaped domain to the optimal domain for Example 5.4 with the improved formula (first row: proposed Algorithm 1 for Stokes flow; second row: Algorithm 1 with volumetric CT-H(sym) gradient flow for Stokes flow; third row: proposed Algorithm 1 for Navier-Stokes flow).

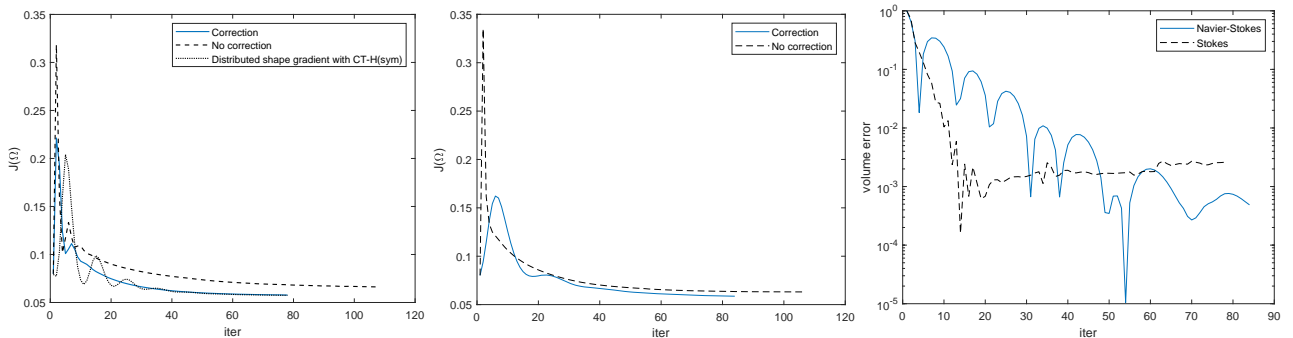


FIG. 5.13. Comparisons on the convergence history between the algorithms with the improved and classical boundary formulae as well as distributed CT-H(sym) gradient flows [26] for Example 5.4: flows of Stokes (left), Navier-Stokes (middle), volume error (right).

**5.1.4. Interface identification.** In this subsection we consider the interface identification problem.

EXAMPLE 5.8. For the elliptic interface problem with  $\sigma_1 = 1$  and  $\sigma_2 = 0.1$ , we assume that  $\Omega = (-1, 1)^2$ ,  $\Omega_1 \subset\subset \Omega_2$  and  $\Gamma = \partial\Omega_1$  (cf. Fig. 3.1), and consider the inverse problem

$$\min_{\Omega} J(\Omega) = \frac{1}{2} \int_{\Omega} (u - u_d)^2 dx.$$

Set  $f = \chi_{\Omega_*}$  with  $\Omega_*$  being a disk of radius 0.5 and centered at (0,0). The observed function  $u_d$  is obtained by solving the state equation on  $\Omega_*$ . Set  $\varepsilon = 10^{-4}$  in Algorithm 1. We see from Fig. 5.21 that the shape gradient algorithm with improved boundary formula converges to the target optimal domain from the square, while the algorithm with classical boundary formula falls into the local minimum thus fails to converge to the optimal shape. Fig. 5.22 shows that the shape gradient algorithm with improved boundary formula converges more efficiently than the algorithm with the classical boundary formula, and reaches a smaller objective value.

**5.1.5. Stokes eigenvalue optimization.** In this subsection we consider the Stokes eigenvalue optimizations.

EXAMPLE 5.9. Set  $C = 1$  and the initial shape to be a unit square (2D) or a unit cube (3D). In 2D, we set  $\beta = 10$ ,  $\beta = 20$ , and  $\beta = 50$  for  $\ell = 1, 2, 3$ ,  $\ell = 4, 5, 6, 7$ , and  $\ell = 8, 9, 10$ , respectively. For 3D, set  $\beta = 100$  for all  $\ell$ . We refer to Table 5.9 (right) for the computed eigenvalues and the corresponding eigen-velocities by using the moving grid method (cf. [3] for comparisons). The convergence histories of the different eigenvalues in 2D are presented in Fig. 5.23. We refer to Table 5.6 for the optimal eigenvalues and the corresponding *new* optimal shapes in 3D. To the best of our knowledge, the optimal shapes of Stokes eigenvalue problems in 3D were not reported in the literature. Table 5.5 shows that the improved boundary formula is faster than the volume type and is expected to be slower than the standard boundary type formula.

**5.2. Fourier parametrization based shape optimization algorithm.** In this subsection we consider the shape optimization algorithm based on the boundary parametrization. In such case, we are led to finite-dimensional optimization problems so that the classical gradient descent algorithms can be used for boundary evolution. Let us first

Formula type	Example 5.4	Example 5.6
Improved Boundary	12.693	30.264
Improved Boundary+Decomposition	10.898	27.502
CT-H(sym)+Volume [26]	12.217	37.512

TABLE 5.4

Comparisons on computational costs (seconds) for the Stokes case of Example 5.4 and Example 5.6 by Algorithm 1 among different shape gradient formulae: 374 and 4556 nodes for the initial L-shape domain and initial rectangle removed by a round disk, respectively.

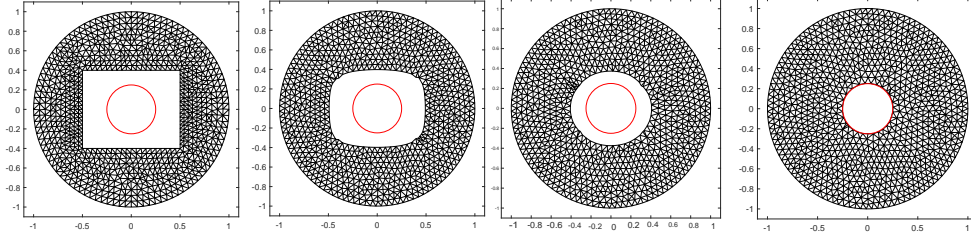


FIG. 5.14. Deformations from an inner square to a circle for Example 5.5 by Algorithm 1.

introduce the Fourier boundary parametrization method [7, 33]. We will consider the class of star-shaped domains in plane. Let  $\Omega$  be  $C^2$  and consider the boundary parametrization

$$\Omega = \{r(\cos \theta, \sin \theta) : \theta \in [0, 2\pi), 0 \leq r < \rho(\theta)\},$$

where  $\rho(\theta)$  is a function. The boundary  $\partial\Omega$  can be parameterized as

$$\partial\Omega = \{\rho(\theta)(\cos \theta, \sin \theta), \theta \in [0, 2\pi)\}. \quad (5.4)$$

For non star-shaped domains, other kinds of parametrization for curve boundaries such as splines or NURBS could be used. By Fourier series expansion for  $\rho(\theta)$ , we have

$$\rho(\theta) := a_0 + \sum_{n=1}^{\infty} a_n \cos(n\theta) + \sum_{n=1}^{\infty} b_n \sin(n\theta), \quad (5.5)$$

which can be approximated by truncated Fourier series:

$$\rho(\theta) \approx \rho_N(\theta) := a_0 + \sum_{n=1}^N a_n \cos(n\theta) + \sum_{n=1}^N b_n \sin(n\theta), \quad (5.6)$$

where the expansion coefficients  $a_n$  and  $b_n$  are to be determined. Then each vector of Fourier coefficients  $(a_0, a_1, \dots, a_N, b_1, b_2, \dots, b_N)$  defines the boundary of a domain. The optimization problem can be solved by searching for an optimal coefficient vector. Then we can obtain formulae for the derivatives of the shape functional in terms of Fourier coefficients. The shape optimization algorithm based on the Fourier parametrization is given in Algorithm 2, where it stops by setting a prescribed iteration number. In the following we take the Dirichlet Laplacian and Stokes eigenvalue optimization as examples. Choose the truncation number  $N = 20$  and a fixed step size 0.001 to update the coefficients.

---

**Algorithm 2:** Fourier parametrization based shape optimization algorithm

---

Choose initial Fourier coefficients for  $\partial\Omega$ ;  
**while** the iteration does not stop **do**  
    Mesh generation;  
    Solve the state problem;  
    Solve the possible adjoint problem;  
    Calculate the outward normal derivatives of the state and adjoint state;  
    Update the Fourier coefficients;  
     $k \leftarrow k + 1$ ;  
**end**

---

**5.2.1. Laplace eigenvalue optimization.** For problem (3.5), the Eulerian derivative of  $J(\Omega) = \text{Vol}(\Omega)\lambda_\ell$  is

$$dJ(\Omega; \mathbf{V}) = \int_{\partial\Omega} [\lambda_\ell - \text{Vol}(\Omega) (\partial_{\mathbf{n}} u_\ell)^2] \mathbf{V}_n ds. \quad (5.7)$$

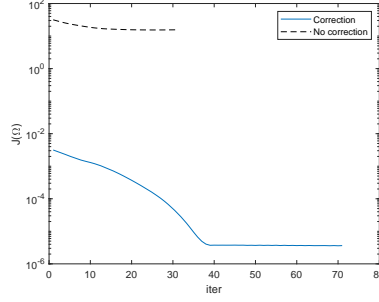


FIG. 5.15. Comparisons on the convergence history of the shape functional between the shape gradient algorithms with improved and classical boundary formulae for Example 5.5.

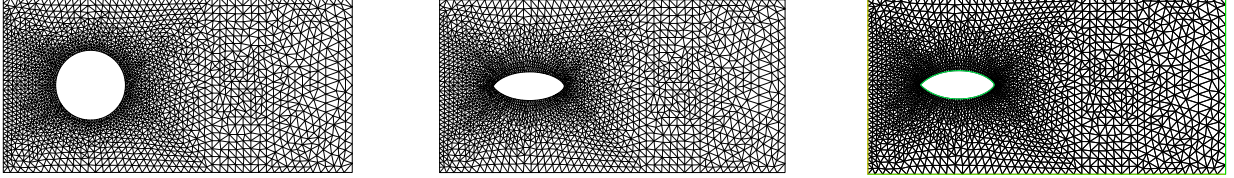


FIG. 5.16. The initial domain (left), the optimal shape with Algorithm 1 (middle), and the optimal shape by the algorithm with a volumetric CT-H(sym) gradient flow [26] (right) for the Stokes case of Example 5.6 :  $\mu = 0.01$ .

On  $\partial\Omega$ , we have by the Fourier parametrization that  $\mathbf{x} = \rho_N(\theta)\mathbf{r}$ , where  $\mathbf{x} = (x_1, x_2)$  and  $\mathbf{r} = [\cos \theta, \sin \theta]^T = \frac{\mathbf{x}}{|\mathbf{x}|}$ . For boundary variations from  $\partial\Omega$  to  $\partial\Omega_t$  due to perturbations of the coefficient  $a_0$ , we denote  $\mathbf{x}_t = \rho_{N,t}(\theta)\mathbf{r}$  on  $\partial\Omega_t$ , where

$$\rho_{N,t}(\theta) := a_0 + t + \sum_{n=1}^N a_n \cos(n\theta) + \sum_{n=1}^N b_n \sin(n\theta).$$

The variations of coefficients induce shape variations of a domain. Denote

$$\Omega_t = \{r\mathbf{r} : \theta \in [0, 2\pi], 0 \leq r < \rho_{N,t}(\theta)\}.$$

Then

$$\mathbf{v} = \left. \frac{d}{dt} \mathbf{x}_t \right|_{t=0} = \left. \frac{d}{dt} \rho_{N,t}(\theta) \right|_{t=0} \mathbf{r} = \mathbf{r}.$$

Then (5.7) implies that the sensitivity of  $J$  with respect to  $a_0$ :

$$\frac{\partial J}{\partial a_0} = \lim_{t \searrow 0} \frac{J(\Omega_t) - J(\Omega)}{t} = dJ(\Omega; \mathbf{v}) = \int_{\partial\Omega} \vartheta^L \mathbf{r} \cdot \mathbf{n} ds, \quad (5.8)$$

where  $\vartheta^L = \lambda_\ell - \text{Vol}(\Omega) (\partial_{\mathbf{n}} u_\ell)^2$ . A similar procedure applied to the coefficients  $a_i$  or  $b_i$  ( $i = 1, \dots, N$ ), we have

$$\begin{aligned} \frac{\partial J}{\partial a_n} &= \int_{\partial\Omega} \vartheta \cos n\theta \mathbf{r} \cdot \mathbf{n} ds, \quad n = 1, 2, \dots, N, \\ \frac{\partial J}{\partial b_n} &= \int_{\partial\Omega} \vartheta \sin n\theta \mathbf{r} \cdot \mathbf{n} ds, \quad n = 1, 2, \dots, N, \end{aligned} \quad (5.9)$$

where  $\theta = \arctan \frac{x_2}{x_1}$ .

On the discrete level, the boundary correction (3.8) is used for the outward normal derivative of the eigenfunction  $u_\ell$  on  $\partial\Omega$ . The discrete formula can be obtained by replacing  $\lambda_\ell$  and  $\partial_{\mathbf{n}} u_\ell$  in (5.8)-(5.9) by  $\lambda_{\ell,h}$  and  $\partial_{\mathbf{n}}^h u_{\ell,h}$ , respectively. In Table 5.7 and Fig. 5.24, we present the computed optimal Laplace eigenvalues, the optimal shapes, and the corresponding convergence histories of different eigenvalues. A comparison with Table 5.2 shows the similar results. In Table 5.8, we compare the computational costs with the standard and improved boundary formulae.

**5.2.2. Stokes eigenvalue optimization.** For problem (3.13), the Eulerian derivative of the Lagrangian is (3.16). By using a quite similar derivation procedure as in Section 5.2.1, we obtain the following discrete formulae:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial a_0} &= \int_{\partial\Omega} \vartheta^S \mathbf{r} \cdot \mathbf{n} ds, \\ \frac{\partial \mathcal{L}}{\partial a_n} &= \int_{\partial\Omega} \vartheta^S \cos n\theta \mathbf{r} \cdot \mathbf{n} ds, \quad n = 1, 2, \dots, N, \\ \frac{\partial \mathcal{L}}{\partial b_n} &= \int_{\partial\Omega} \vartheta^S \sin n\theta \mathbf{r} \cdot \mathbf{n} ds, \quad n = 1, 2, \dots, N, \end{aligned}$$

where  $\vartheta^S = l + \beta(\text{Vol}(\Omega) - C) - \|\mathbf{D}_{\mathbf{n}}^h \mathbf{u}_h\|^2$  with the modified boundary formula (3.18) being used. The numerical results on optimal shapes are summarized in Table 5.9 (left), where a comparison with the results obtained by moving grid based shape optimization algorithm can be performed.



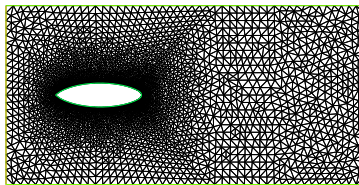


FIG. 5.17. The optimal shape of the Navier-Stokes case with  $\mu = 0.1$  for Example 5.6 with Algorithm 1.

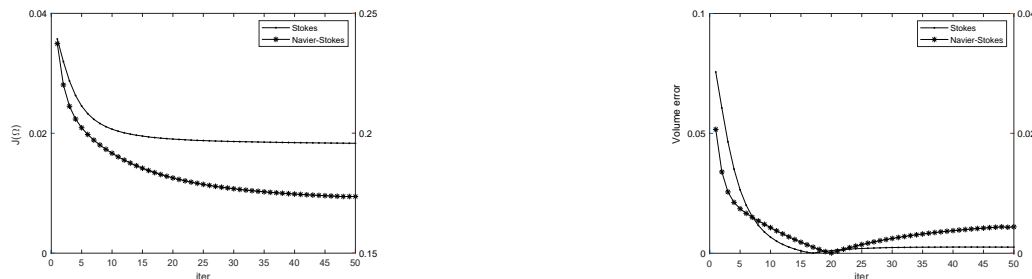


FIG. 5.18. Convergence histories of the objective (left) and the volume error (right) for Example 5.6 with Algorithm 1.

**6. Conclusion.** The continuity property of the discrete boundary shape gradient is helpful in certain shape optimization algorithms and provides certain flexibility compared to the previous widely-used discontinuous ones and the distributed shape gradients in the literature. We have proposed modified continuous shape gradient formulae of boundary type based on the discrete variational outward normal derivatives for various shape optimizations of Dirichlet problems, including the eigenvalue optimization, the optimal design in fluid flows, and the interface identification problems. Numerical accuracy was verified in different shape optimization problems and numerical performance was investigated in several popular shape optimization algorithms. Moreover, a continuous approximate shape gradient of boundary type for Neumann problems has been proposed by using gradient recovery techniques. The proposed two modified boundary type shape gradients can also be used in topology optimizations in combination with, e.g., the level set method, which will be reported elsewhere.

**Acknowledgements.** The authors would like to thank two anonymous referees for their careful reading of the manuscript and many valuable suggestions, which helped to improve the presentation and quality of this paper greatly.

## REFERENCES

- [1] G. Allaire and A. Henrot, On some recent advances in shape optimization, *C. R. Acad. Sci. Paris, Ser. II*, 329 (2001), pp. 383-396.
- [2] G. Allaire, C. Dapogny, and F. Jouve, Shape and topology optimization, in *Geometric partial differential equations, part II*, A. Bonito and R. Nochetto eds., pp. 1-132, *Handbook of Numerical Analysis*, vol. 22, Elsevier, 2021.
- [3] P. Antunes, Optimal bilaplacian eigenvalues, *SIAM J. Control Optim.*, 52 (2014), pp. 2250-2260.
- [4] P.R.S. Antunes and P. Freitas, Numerical optimization of low eigenvalues of the Dirichlet and Neumann Laplacians, *J. Optim. Theory Appl.*, 154 (2012), pp. 235-257.
- [5] M.P. Bendsøe and O. Sigmund, *Topology Optimization. Theory, Methods and Applications*, Springer-Verlag, Berlin, 2003.
- [6] M. Berggren, A unified discrete-continuous sensitivity analysis method for shape optimization. In: *Applied and Numerical Partial Differential Equations*, pp. 25-39, Springer, New York, 2010.
- [7] B. Bogosel, *Shape Optimization and Spectral Problems*, PhD Thesis, 2015.
- [8] C. Brandenburg, F. Lindemann, M. Ulbrich and S. Ulbrich, A continuous adjoint approach to shape optimization for Navier-Stokes flow, *Optimal control of coupled systems of partial differential equations*, pp. 35-56, *Internat. Ser. Numer. Math.*, 158, Birkhäuser Verlag, Basel, 2009.
- [9] S.C. Brenner and L.R. Scott, *The Mathematical Theory of Finite Element Methods*, 3rd edn., Springer, New York, 2008.
- [10] R. Brügger, H. Harbrecht, and J. Tausch, On the numerical solution of a time-dependent shape optimization problem for the heat equation, *SIAM J. Control Optim.*, 59 (2021), pp. 931-953.
- [11] M. Burger, A framework for the construction of level set methods for shape optimization and reconstruction, *Interfaces Free Bound.*, 5 (2003), pp. 301-329.
- [12] T. Belytschko, S.P. Xiao, and C. Parimi, Topology optimization with implicit functions and regularization, *Int. J. Numer. Methods Engrg.*, 57 (2003), pp. 1177-1196.
- [13] J. Céa, Conception optimale ou identification de formes, calcul rapide de la dérivée directionnelle de la fonction coût, *ESAIM: Math. Model. Numer. Anal.*, 20 (1986), pp. 371-402.
- [14] Y. Chitour, D. Kateb, and R. Long, Generic properties of the spectrum of the Stoke system with Dirichlet boundary condition in  $\mathbb{R}^3$ , *Ann. I.H.Poincaré*, 33 (2016), pp. 119-167.
- [15] M.C. Delfour and J.P. Zolésio, *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization*, 2nd edn., SIAM, Philadelphia, 2011.
- [16] P. Gangl, K. Sturm, M. Neunteufel and J. Schöberl, Fully and semi-automated shape differentiation in ngsolve, *Struct. Multidiscip. Optim.*, 63 (2021), pp. 1579-1607.
- [17] W. Gong and S. Zhu, On discrete shape gradients of boundary type for PDE-constrained shape optimization, *SIAM J. Numer. Anal.*, 59 (2021), pp. 15101541.

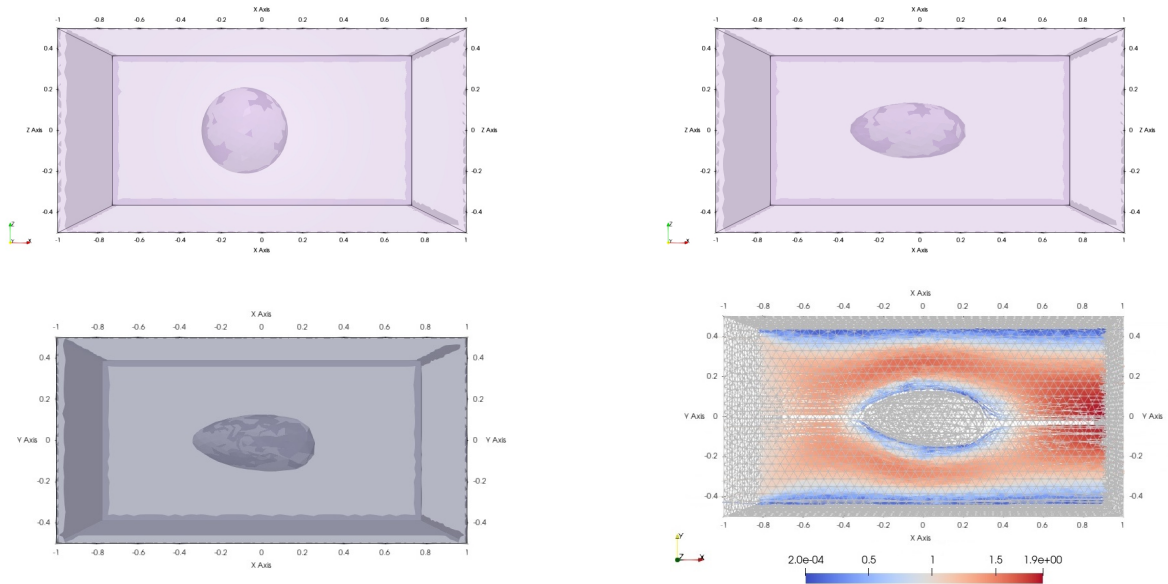


FIG. 5.19. The initial design (top left), the computed optimal shape of Stokes case with  $\mu = 0.1$  (top right), and the computed optimal shape of Navier-Stokes case (bottom left) with  $\mu = 0.01$  and its velocity (bottom right) for Example 5.7.

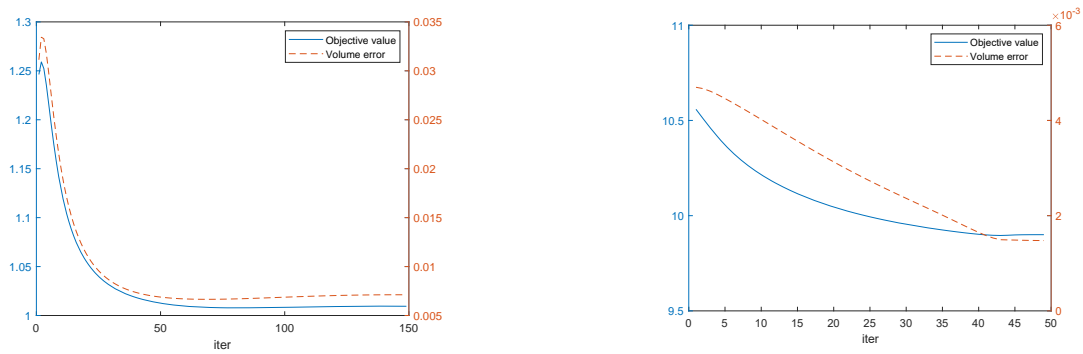


FIG. 5.20. Convergence histories of the objective and volume error for Example 5.7: Stokes with  $\mu = 0.01$  (left) and Navier-Stokes with  $\mu = 0.1$  (right).

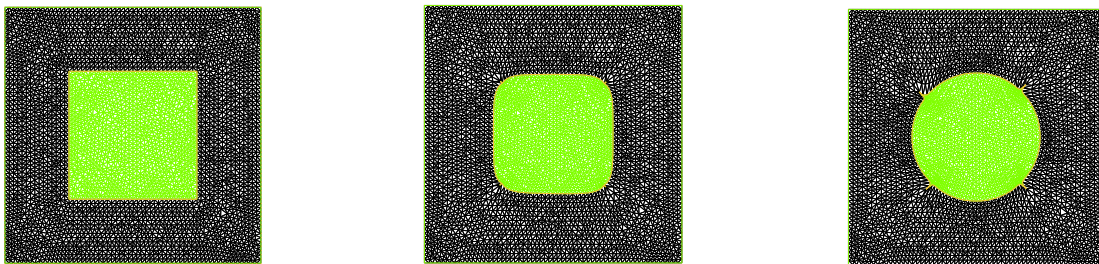


FIG. 5.21. The initial design (left), the optimal shape by classical boundary shape gradient (middle) and the optimal shape using improved boundary formula (right) for Example 5.8.

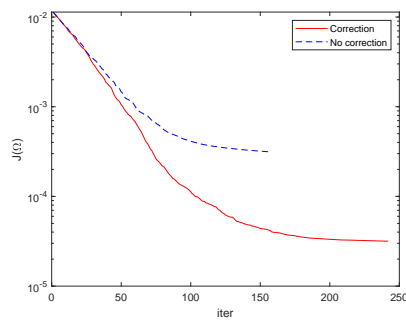


FIG. 5.22. Convergence history of the shape functional by shape gradient algorithms with the improved and classical boundary formulae for Example 5.8.

Formula type	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	$\lambda_5$	$\lambda_6$	$\lambda_7$	$\lambda_8$	$\lambda_9$	$\lambda_{10}$
Volume	466.13	513.65	481.82	502.88	464.07	476.17	457.85	458.42	496.66	438.39
Boundary	422.19	397.25	432.14	363.40	384.69	393.11	410.49	408.13	406.52	364.35
Improved Boundary +Decomposition	395.40	400.71	403.34	387.96	407.49	408.43	404.80	410.05	390.49	396.33

TABLE 5.5

Comparisons on computational costs (seconds) for Example 5.9 (2D) by Algorithm 1 associated with different  $H^1$  gradient flows.

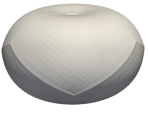
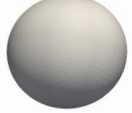
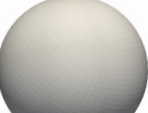

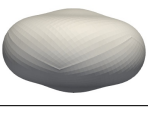

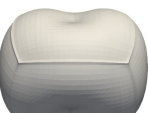

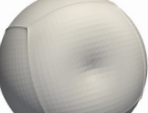

$\ell$	$m$	$\lambda_\ell$	Optimal shape	$\ell$	$m$	$\lambda_\ell$	Optimal shape
1	1	49.560		2	1	52.797	
3	3	53.082		4	2	69.561	
5	2	74.684		6	2	81.367	
7	3	83.495		8	5	85.812	
9	6	86.762		10	7	87.808	

TABLE 5.6

Numerical results for Example 5.9 on the Stokes eigenvalue optimization in 3D.

- [18] J. Hadamard, Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées, in Mémoires présentés par divers savants à l'Académie des Sciences, vol. 33, Imprimerie nationale, 1908.
- [19] H. Harbrecht and J. Tausch, On the numerical solution of a shape optimization problem for the heat equation, SIAM J. Sci. Comput., 35 (2013), pp. A104-A121.
- [20] J. Haslinger and R. Mäkinen, Introduction to Shape Optimization: Theory, Approximation, and Computation, SIAM, 2003.
- [21] J. Haslinger and P. Neittaanmaki, Finite Element Approximation for Optimal Shape, Material and Topology Design, 2nd edition, J. Wiley & Sons: Chichester, 1996.
- [22] E.J. Haug, K.K. Choi and V. Komkov, Design Sensitivity Analysis of Structural Systems, Academic Press, Orlando, FL, 1986.
- [23] A. Henrot, Extremum Problems for Eigenvalues of Elliptic Operators. Frontiers in Mathematics, Birkhäuser, Basel, 2006.
- [24] R. Hiptmair, A. Paganini, and S. Sargheini, Comparison of approximate shape gradients, BIT Numer. Math., 55 (2015), pp. 459-485.
- [25] F. Hecht, New development in FreeFem++, J. Numer. Math., 20 (2012), pp. 251-265.
- [26] J.A. Iglesias, K. Sturm and F. Wechsung, Two-dimensional shape optimization with nearly conformal transformations, SIAM J. Sci. Comput., 40 (2018), pp. A3807-A3830.
- [27] J. P. Kelliher, Eigenvalues of the Stokes operator versus the Dirichlet Laplacian in the plane, Pac. J. Math., 244 (2010), pp. 99-132.
- [28] A. Laurain and K. Sturm, Distributed shape derivative via averaged adjoint method and applications, ESAIM: Math. Model. Numer. Anal., 50 (2016), pp. 1241-1267.
- [29] J. Li and S. Zhu, Shape identification in Stokes flow with distributed shape gradients, Appl. Math. Lett., 95 (2019), pp. 165-171.
- [30] B. Mohammadi and O. Pironneau, Applied Shape Optimization for Fluids, 2nd ed., Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 2010.
- [31] S. Osher and J. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys., 79 (1988), pp. 12-49.
- [32] E. Oudet, Numerical minimization of eigenmodes of a membrane with respect to the domain, ESAIM: Control Optim. Calc. Var., 10 (2004), pp. 315-330.
- [33] B. Osting, Optimization of spectral functions of DirichletLaplacian eigenvalues, J. Comput. Phys., 229 (2010), pp. 8578-8590.
- [34] J.H. Ortega and E. Zuazua, Generic simplicity of the eigenvalues of the Stokes system in two space dimensions, Adv. Differ. Equ., 6 (2001), pp. 987-1023.
- [35] A. Paganini, Approximate shape gradients for interface problems, New Trends in Shape Optimization, A. Pratelli and Leugering (eds.), International Series of Numerical Mathematics 166, 217-227, 2015.

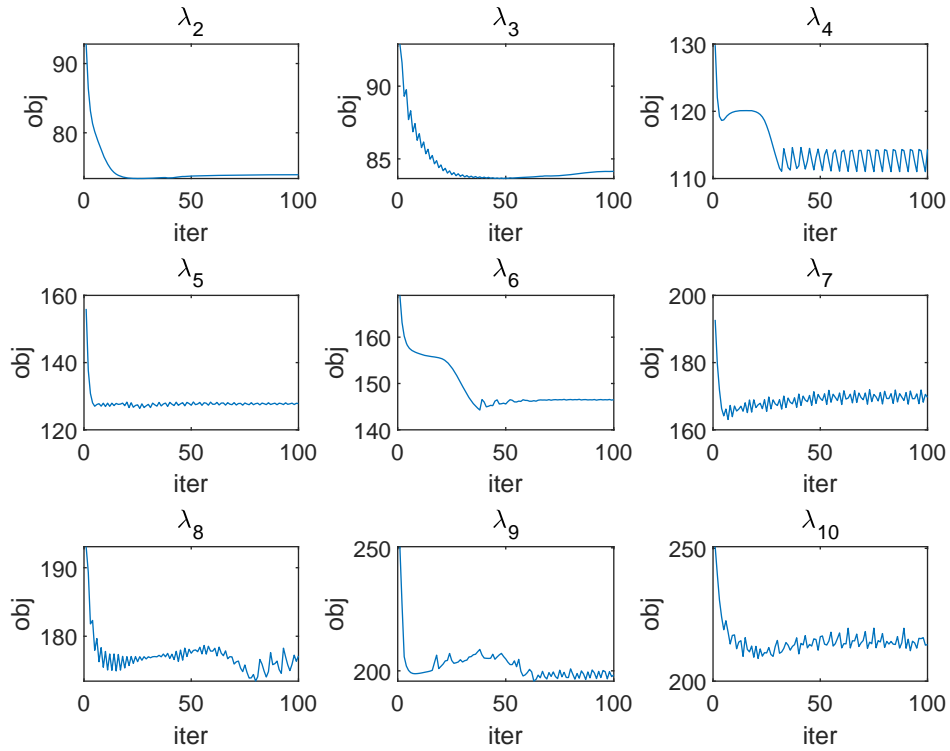


FIG. 5.23. Convergence histories of the Stokes eigenvalue optimization by using Algorithm 1 for Example 5.9 in 2D ( $\ell = 2, 3, \dots, 10$ ).

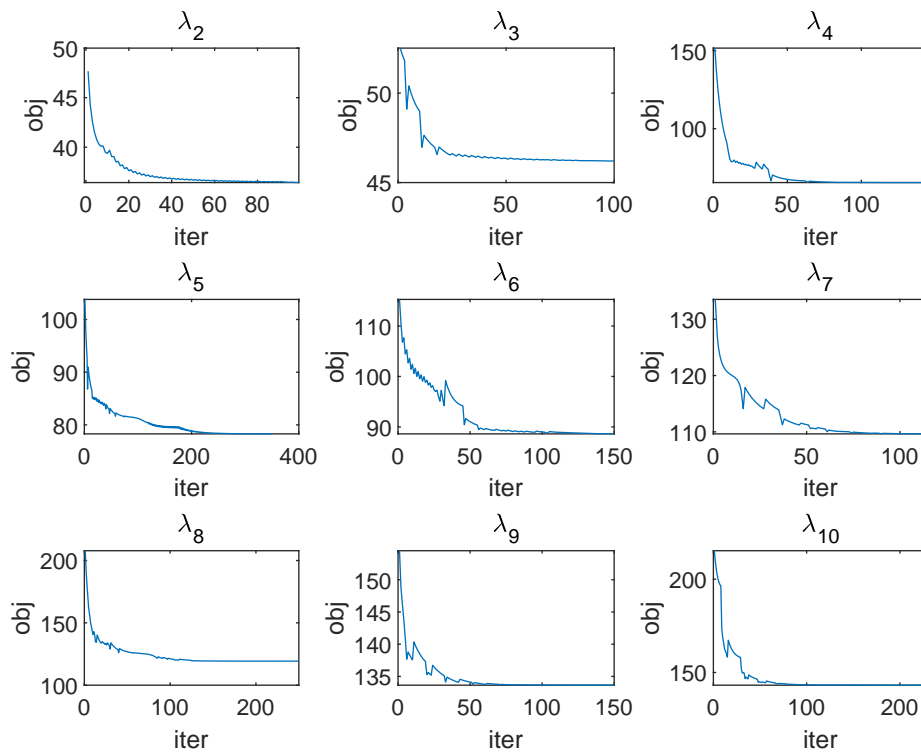


FIG. 5.24. Convergence history of the Laplacian eigenvalue optimization by Algorithm 2 for Example 5.3 ( $\ell = 2, 3, \dots, 10$ ).

- [36] A. Paganini and F. Wechsung, Fireshape: a shape optimization toolbox for firedrake, Struct. Multidiscip. Optim., 63 (2021), pp. 2553-2569.
- [37] O. Pironneau, Optimal Shape Design for Elliptic Systems, Springer, New York, 1984.
- [38] V.H. Schulz and M. Siebenborn, Computational comparison of surface metrics for PDE constrained shape optimization, Comput. Methods Appl. Math., 16 (2016), pp. 485-496.
- [39] V.H. Schulz, M. Siebenborn, and K. Welker, Structured inverse modeling in parabolic diffusion problems, SIAM J. Control Optim., 53 (2015), pp. 3319-3338.
- [40] 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.











$\ell$	$m$	$\lambda_\ell$	Optimal shape	$\ell$	$m$	$\lambda_\ell$	Optimal shape
1	1	18.175		2	1	36.386	
3	2	46.175		4	2	65.749	
5	2	78.294		6	3	88.616	
7	3	109.643		8	3	119.294	
9	3	133.641		10	4	143.215	

TABLE 5.7

Numerical results for Example 5.3 on the Laplace eigenvalue optimization with Fourier parametrization based algorithm.

Formula type	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	$\lambda_5$	$\lambda_6$	$\lambda_7$	$\lambda_8$	$\lambda_9$	$\lambda_{10}$
Boundary	21.92	22.89	20.78	21.09	20.44	20.43	20.00	24.58	25.55	20.35
Improved Boundary	27.32	25.40	26.53	26.67	25.95	25.66	26.02	26.64	28.67	26.90

TABLE 5.8

Comparisons on computational costs (seconds) for Example 5.3 (2D) by Algorithm 2 with the standard and modified boundary formulae.

- [41] J. Sokolowski and J.P. Zolésio, Introduction to Shape Optimization: Shape Sensitivity Analysis, Springer, Heidelberg, 1992.
- [42] M.Y. Wang, X. Wang, and D. Guo, A level-set method for structural topology optimization, Comput. Meth. Appl. Mech. Engrg., 192 (2003), pp. 227-246.
- [43] W. Yan and Y. Ma, Shape reconstruction of an inverse Stokes problem, J. Comput. Appl. Math., 216 (2008), pp. 554-562.
- [44] N.Y. Yi, Y.Q. Huang, and W. Yang, Function, derivative and high-order derivatives recovery methods using the local symmetry projection, J. Sci. Comput., 74 (2018), pp. 536-572.
- [45] S. Zhu, Effective shape optimization of Laplace eigenvalue problems using domain expressions of Eulerian derivatives, J. Optim. Theory Appl., 176 (2018), pp. 17-34.
- [46] S. Zhu and Z. Gao, Convergence analysis of mixed finite element approximations to shape gradients in the Stokes equation, Comput. Methods Appl. Mech. Engrg., 343 (2019), pp. 127-150.
- [47] S. Zhu, X. Hu and Q. Liao, Convergence analysis of Galerkin finite element approximations to shape gradients in eigenvalue optimization, BIT Numer. Math., 60 (2020), pp. 853-878.
- [48] O.C. Zienkiewicz and J.Z. Zhu, The superconvergent patch recovery and a posteriori error estimates, Int. J. Numer. Methods Eng., 33 (1992), pp. 1331-1382.

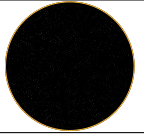
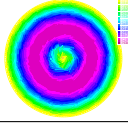
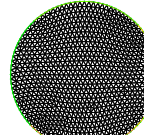
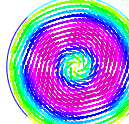

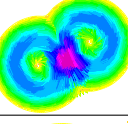
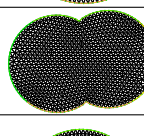
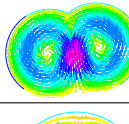
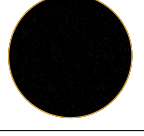
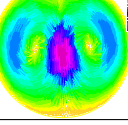
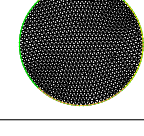
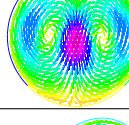

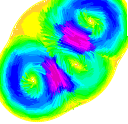
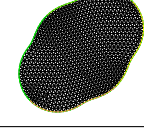
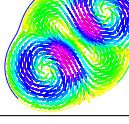
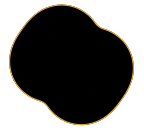
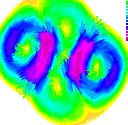
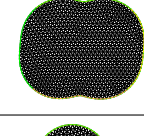
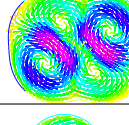

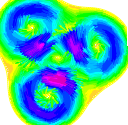
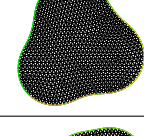
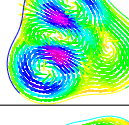

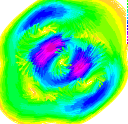
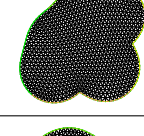
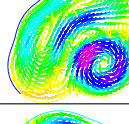

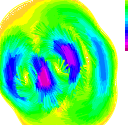
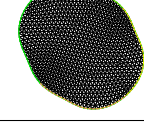
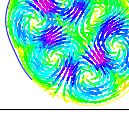

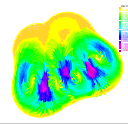
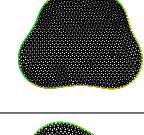
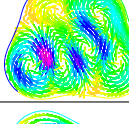
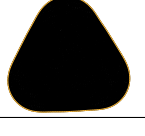
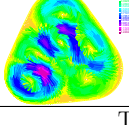
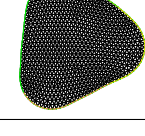
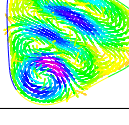
$\ell$	2D				$\ell$	2D			
	$m$	$\lambda_\ell$	Optimal shape	Velocity field		$m$	$\lambda_\ell$	Optimal shape	Velocity field
1	1	46.163			1	1	46.537		
2	1	73.282			2	1	74.049		
3	2	83.028			3	2	84.169		
4	1	110.392			4	1	112.578		
5	2	125.484			5	2	127.541		
6	3	143.017			6	3	146.560		
7	2	168.234			7	2	171.605		
8	3	175.602			8	3	178.011		
9	3	195.846			9	3	197.237		
10	4	208.881			10	4	212.702		

TABLE 5.9

Numerical results on the Stokes eigenvalue optimization in 2D with the Fourier parametrization (left) and moving grid (right) based algorithms.