Raino A. E. Mäkinen

Tuomo Rossi

Jari Toivanen

A moving mesh fictitious domain approach for shape optimization problems

A MOVING MESH FICTITIOUS DOMAIN APPROACH FOR SHAPE OPTIMIZATION PROBLEMS

RAINO A.E. MÄKINEN[1], TUOMO ROSSI[1] AND JARI TOIVANEN[1]

Abstract. A new numerical method based on fictitious domain methods for shape optimization problems governed by the Poisson equation is proposed. The basic idea is to combine the boundary variation technique, in which the mesh is moving during the optimization, and efficient fictitious domain preconditioning in the solution of the (adjoint) state equations. Neumann boundary value problems are solved using an algebraic fictitious domain method. A mixed formulation based on boundary Lagrange multipliers is used for Dirichlet boundary problems and the resulting saddle-point problems are preconditioned with block diagonal fictitious domain preconditioners. Under given assumptions on the meshes, these preconditioners are shown to be optimal with respect to the condition number. The numerical experiments demonstrate the efficiency of the proposed approaches.

1. INTRODUCTION

The basic idea of fictitious domain (domain embedding) methods is to extend a partial differential equation given in a complex shaped domain to a larger but simpler domain such as a rectangle or a parallelepiped. Also, the capacitance matrix methods have the same idea, but slightly different implementation; see, for example, [38] and references therein. The solution of the discretized and extended partial differential equation is obtained very efficiently using the preconditioned iterative methods. The preconditioning step can be realized using the fast direct methods [42]. The extension of the original problem is made such that the solution is obtained as a restriction. This can be accomplished in several ways. The following functional analytic fictitious domain methods have been studied mostly by the French school led by Glowinski: In the optimal control approach [2,27], the problem is formulated as a distributed optimal control problem. Another possibility is to add some constraints to the extended problem. This can be done using boundary Lagrange multipliers [18,19,44] or using distributed Lagrange multipliers [16,17,27,44].

Opposite to the functional analytic approaches, in the algebraic fictitious domain methods the problem is extended so that the solution of the original problem is obtained directly as a restriction of the extended problem without any additional constraints. In this case, the formulation is typically done at the matrix level. See, for example, [1,4,6,34,39]. With the algebraic approach a lot of attention is given to the efficient solution of the arising linear problem. It is well known [1] that for second order elliptic Neumann boundary value problems,

---

[1] Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (MaE), 40351 Jyväskylä, Finland.
e-mail: `Raino.Makinen@mit.jyu.fi`, `Tuomo.Rossi@mit.jyu.fi`, `Jari.Toivanen@mit.jyu.fi`

the trivial extension by zero and the use of the same operator, which is extended to the whole domain, as preconditioner is optimal with respect to the condition number. We will use this approach for Neumann boundary value problems.

It is more difficult to construct an optimal extension and preconditioner for Dirichlet boundary values problems. Here, we have chosen to use the mixed formulation in which the problem is formulated as a Neumann boundary value problem with boundary Lagrange multipliers; see [3, 8]. For the saddle-point problem obtained in this way, Kuznetsov introduced a block diagonal fictitious domain preconditioner which is optimal for the zero extended Neumann problem with respect to the condition number [31, 39]. For domain decomposition methods a similar approach has been considered in [30, 32]. Another possibility would be to use distributed Lagrange multipliers, especially the approach considered in [17] which is well suited also for three-dimensional problems and mixed boundary value problems.

It is advantageous to use the fictitious domain methods with a fixed mesh especially when solving the state equation in shape optimization problems as well as in other moving boundary problems. The computation of the stiffness matrices and load vectors for different designs is usually very simple unlike in many of the other approaches and the solution procedure is often efficient even though the domain is changing. However, there are some drawbacks when the mesh is kept fully rectangular during the optimization. It may lead to nonsmooth objective functions, locking effects and low approximation accuracy; see, for example, the review paper [22] by Haslinger. In the shape optimization problems considered in [18,25,29], the fictitious domain method enforces the Dirichlet boundary value in the state equation using Lagrange multipliers on the boundary. In [13,21,23,24], a distributed control is introduced to the right-hand side and the Dirichlet boundary value is added to the objective function as a penalty. In [12], the Neumann boundary condition is formulated as a constraint using Lagrange multipliers on the boundary. In [16,44], the use of distributed Lagrange multipliers is considered.

Unlike usually, we will use meshes which are fitted to the boundaries and they are moving during the optimization. Hence, the considered method is a combination of the boundary variation technique and the fictitious domain method. Some of the advantages of the fixed mesh fictitious domain approach are lost due to this, namely, the mesh generation is a bit more complicated and the discrete state equation must be formed in each iteration. The important aspect is that the state and adjoint state equations can be very efficiently solved although the mesh is moving. Also, by choosing this approach, we get rid of the typical problems with the fixed meshes such as nonsmooth objective functions, locking effects and low approximation accuracy. The moving mesh approach was used in [43] for solving shape optimization problems with subsonic full potential flows.

The rest of this paper is arranged as follows: First, the shape optimization problems for Dirichlet and Neumann boundary value problems are formulated. Next, the parametrization of shape, the mesh generation and the finite element discretization of the state equation are considered. Then, the sensitivity analysis with the adjoint state technique is introduced and efficient solution procedures for the arising linear systems are constructed. In the last section, the numerical experiments demonstrate the efficiency and the robustness of the proposed methods.

2. FORMULATION OF THE SHAPE OPTIMIZATION PROBLEMS

Let Ω_{in} and Ω_{out} be given non-empty bounded domains in $\mathbb{R}^2$ such that $\Omega_{\text{in}} \subseteq \Omega_{\text{out}}$. The class of admissible domains Ω is defined by

$$\mathcal{O} = \big\{ \Omega \subset \mathbb{R}^2 \mid \Omega_{\text{in}} \subseteq \Omega \subseteq \Omega_{\text{out}}, \ \Omega \text{ is open and simply connected,}$$
$$\Omega \text{ has } \varepsilon\text{-cone property} \big\} . \tag{1}$$

For any $\Omega \in \mathcal{O}$, the state equation in the case of a Neumann boundary value problem reads:

$$\begin{cases} -\Delta u + cu = f & \text{in } \Omega, \\ \dfrac{\partial u}{\partial n} = g & \text{on } \partial\Omega, \end{cases} \tag{2}$$

or in weak form

$$\begin{cases} \text{Find } u \in H^1(\Omega) \text{ such that} \\ \displaystyle\int_\Omega (\nabla u \cdot \nabla v + c\,u\,v)\,\mathrm{d}x = \int_\Omega f\,v\,\mathrm{d}x + \int_{\partial\Omega} g\,v\,\mathrm{d}s \quad \forall v \in H^1(\Omega), \end{cases} \qquad (3)$$

where $f \in L_2(\Omega)$, $g \in L_2(\partial\Omega)$, and $c \geq 0$ is a constant. In the case $c = 0$, we assume for the existence of solution the compatibility condition

$$\int_\Omega f\,\mathrm{d}x + \int_{\partial\Omega} g\,\mathrm{d}s = 0. \qquad (4)$$

For any $\Omega \in \mathcal{O}$, the state equation in the case of a Dirichlet boundary value problem reads:

$$\begin{cases} -\Delta u + cu = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \qquad (5)$$

or in weak form

$$\begin{cases} \text{Find } u \in H_0^1(\Omega) \text{ such that} \\ \displaystyle\int_\Omega (\nabla u \cdot \nabla v + c\,u\,v)\,\mathrm{d}x = \int_\Omega f\,v\,\mathrm{d}x \quad \forall v \in H_0^1(\Omega), \end{cases} \qquad (6)$$

where again $f \in L_2(\Omega)$ and $c \geq 0$ is a constant.

Let us assume that $\mathcal{O}$ is equipped with an appropriate topology. Let $J : \mathcal{O} \times H^1(\Omega_{\text{out}}) \to \mathbb{R}$ be a continuous cost functional. Then, the optimal shape design problem reads

$$\min_{\Omega \in \mathcal{O}} J(\Omega, u) \qquad (7)$$

with u being the solution of (3) or (6) depending on the type of the state equation.

According to Haslinger, Neittaanmäki [26], Pironneau [37] and Sokolowski, Zolesio [41], the following theorem holds.

Theorem 2.1. *The optimal shape design problem* (7) *has a solution.*

3. DISCRETIZATION AND SENSITIVITY ANALYSIS

3.1. Parametrization of the shape

In order to solve the shape optimization problem (7) numerically, the boundary of the domain Ω must be parametrized using a finite number of design parameters. Let the vector of design parameters $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)^T \in \mathbb{R}^m$ define the boundary curve. One possible way to perform this is to use the B-spline curves or the Bézier curves; see [7], for example. Then, the design parameters (or variables) are the nodal coordinates of the key points defining the Bézier or B-spline curves. Since the design parameters in $\boldsymbol{\alpha}$ define uniquely the domain Ω_α, in the following, the cost functional is assumed to be of the form $\mathcal{I}(\boldsymbol{\alpha}, u)$ instead of $J(\Omega_\alpha, u)$.

An arbitrary domain Ω must have given geometrical properties in order to be acceptable, that is, it must belong to the set $\mathcal{O}$. Correspondingly, an acceptable design parameter vector $\boldsymbol{\alpha}$ must belong in set of the admissible designs $U_{\text{ad}} \subset \mathbb{R}^m$, which is assumed to be compact and convex.
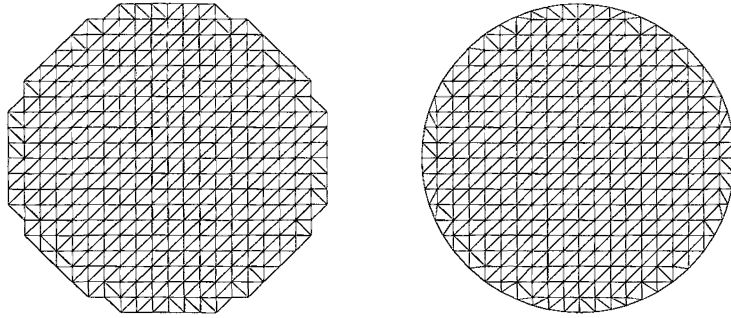
FIGURE 1. An example of triangulations $\mathcal{T}_P$ and $\mathcal{T}_{\Omega_0}$ for a circle.

3.2. Discretization of state equation

In the discretization of the variational problem (3), the standard piecewise linear finite elements are used resulting in a linear system of equations

$$A_\alpha u_\alpha = f_\alpha. \tag{8}$$

Therefore, we must have an algorithm which can triangulate an arbitrary domain Ω in $\mathcal{O}$. In the following, we assume that the characteristic mesh step size h is fixed and, for simplicity, we omit it from our notations. Let us denote the triangulation for the domain Ω by $\mathcal{T}_\Omega = \{\tau_\Omega^k\}_{k=1,\dots,M}$, where τ_Ω^k is the kth triangle in the triangulation $\mathcal{T}_\Omega$. The topological changes in the triangulation during the numerical solution of a shape optimization problem would cause some problems such as the discontinuities in the cost functional of the discretized problem. In order to avoid these difficulties, we shall keep a fixed topology of the triangulation during the solution.

Let Ω_0 be a domain in $\mathcal{O}$, for example, the initial guess for the optimal domain. Let Π be a rectangle such that $\Omega_{\text{out}} \subset \Pi$ and Π is independent of h. Moreover, let there be a rectangular grid covering for Π. The triangulation $\mathcal{T}_\Pi$ is constructed by dividing each cell of this grid into two triangles. This is performed in such a way that a polygonal approximation

$$P = \bigcup_{\tau \in \mathcal{T}_P \subset \mathcal{T}_\Pi} \tau \tag{9}$$

for Ω_0 is obtained. In (9), $\mathcal{T}_P$ is the triangulation for the polygon P. In Figure 1 (a), an example of a triangulation $\mathcal{T}_P$ is shown for a circle.

Let $\psi_\Omega : P \to \bar{\Omega}$ be a homeomorphism, that is, a one-to-one mapping between P and $\bar{\Omega}$ such that both ψ_Ω and its inverse are continuous. Furthermore, let $\psi_{\Omega,\tau} : \tau \to \mathbb{R}^2$ be the linear approximation of ψ_Ω in the triangle $\tau \in \mathcal{T}_P$ which coincides with ψ_Ω in all vertices of τ. Now, the triangulation for the domain Ω is given by $\mathcal{T}_\Omega = \{\psi_{\Omega,\tau}(\tau)\}_{\tau \in \mathcal{T}_P}$. These mappings can be constructed, for example, using an explicit formula, the Laplacian smoothing [14] or combining either of these with a local fitting procedure as in [4,5]. An example of a triangulation $\mathcal{T}_\Omega$ for a circle obtained using the function ψ_Ω constructed in Section 5.1 is shown in Figure 1 (b).

For the regularity of the triangulation obtained in this way, we have the following conditions:

Assumption 3.1. *Let the Jacobian of $\psi_{\Omega,\tau}$ be denoted by $D\psi_{\Omega,\tau}$. There exists a constant c_1 independent of the mesh step size h and the domain $\Omega \in \mathcal{O}$ such that*

$$\max_{\tau \in \mathcal{T}_P} \text{Cond}\left[(D\psi_{\Omega,\tau})^T (D\psi_{\Omega,\tau})\right] \le c_1,$$

where Cond[$\cdot$] *is the spectral condition number. Furthermore, it is assumed that the resulting mesh on the boundary $\partial\Omega$ is quasiuniform, that is, the ratio between the lengths of the longest side and the shortest side of triangle on $\partial\Omega$ is bounded from above by a constant c_2 which is independent of h and Ω.*

For a typical triangulation constructed in this way, the distance between the boundary $\partial\Omega$ and its approximation using the triangulation $\mathcal{T}_\Omega$ is $O(h^2)$ if $\partial\Omega$ in C^2.

3.3. Sensitivity analysis

The discretized shape optimization problem reads now

$$\min_{\alpha\in U_{\mathrm{ad}}} \mathcal{J}(\alpha), \tag{10}$$

where $\mathcal{J}(\alpha) = \mathcal{I}(\alpha, u_\alpha)$ with u_α being the solution of (8). The relation between the solutions of the approximate and continuous shape optimization problems has been extensively studied in [26], for example, and it will not be addressed in this work.

The shape optimization problem (10) can be seen as a constrained minimization problem in $\mathbb{R}^m$. Each cost functional evaluation requires the solution of the state equation which can be a computationally very laborious task. Thus, efficient optimization methods are preferred in order to reduce the number of evaluations of the cost functional. Usually, gradient-based optimization methods are used. For these methods, the cost functional must be smooth enough. For example, the sequential quadratic programming (SQP) algorithm [15] requires that the cost functional is at least twice-continuously differentiable.

In order to obtain the gradient of the cost functional with respect to the design parameters, we perform the sensitivity analysis for an arbitrary design parameter α_k, $1 \leq k \leq m$ using the well-known adjoint equation technique, see, for example, [26]. First, we differentiate the state equation (8) with respect to α_k and obtain

$$A\frac{\partial u}{\partial \alpha_k} = \frac{\partial f}{\partial \alpha_k} - \frac{\partial A}{\partial \alpha_k}u. \tag{11}$$

Next, let q be the solution of the adjoint equation

$$A^T q = \nabla_u \mathcal{I}(\alpha, u), \tag{12}$$

where the right-hand side is the gradient with respect to the state u. Now, the partial derivative is given by

$$\frac{\partial \mathcal{J}(\alpha)}{\partial \alpha_k} = \frac{\partial \mathcal{I}(\alpha, u)}{\partial \alpha_k} + q^T \left(\frac{\partial f}{\partial \alpha_k} - \frac{\partial A}{\partial \alpha_k}u \right). \tag{13}$$

Therefore, we must know how to compute the partial derivatives $\frac{\partial \mathcal{I}}{\partial \alpha_k}(\alpha, u)$, $\frac{\partial f}{\partial \alpha_k}$, $\frac{\partial A}{\partial \alpha_k}$ for all k and the gradient $\nabla_u \mathcal{I}(\alpha, u)$ in order to be able to compute the gradient of $\mathcal{J}$ with respect to the design. All these can be obtained in the standard manner; see, for example, [10, 26, 33, 37, 41]. The adjoint state equation (12) can be solved in the same way as the state equation.

4. A MOVING MESH FICTITIOUS DOMAIN METHOD
FOR ELLIPTIC BOUNDARY VALUE PROBLEMS

4.1. Neumann boundary conditions

For the discretization of the variational form (3), we construct continuous piecewise linear finite element subspaces V_Ω and V_Π approximating $H^1(\Omega)$ and $H^1_0(\Pi)$ using triangulations $\mathcal{T}_\Omega$ and $\mathcal{T}_\Pi$, respectively. By using the Courant basis functions for V_Ω, we obtain for (3) the system of linear equations

$$A_\alpha u_\alpha = f_\alpha. \tag{14}$$

The system (14) is solved using an iterative scheme

$$D(u_\alpha^{k+1} - u_\alpha^k) = f_\alpha - A_\alpha u_\alpha^k, \tag{15}$$

which is accelerated with the CG method. Here, D is an asymptotically optimal preconditioner for any $\Omega \in \mathcal{O}$.

Let us now consider the construction of the preconditioning matrix D. By substituting the Courant basis functions for V_Π into the bilinear form

$$b(u,v) = \int_\Pi (\nabla u \cdot \nabla v + c\,u\,v)\,\mathrm{d}x \qquad u,v \in H_0^1(\Pi), \tag{16}$$

we obtain a matrix having the block form

$$B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \tag{17}$$

where the subscripts 1 and 2 correspond to the nodes in the polygon P and the nodes outside P, respectively. Now, the preconditioning matrix D is the Schur complement matrix

$$D = B_{11} - B_{12}B_{22}^{-1}B_{21}. \tag{18}$$

Remark 4.1. For simplicity, we have chosen to use the Sobolev space $H_0^1(\Pi)$, which corresponds to a Dirichlet boundary value problem, in the construction of the preconditioner D. Another possibility is to use the space $H^1(\Pi)$ corresponding to a Neumann or a Robin boundary value problem. This choice should yield a slightly better preconditioner in terms of the condition number considered in Theorem 4.1; see [6], for example. In any case, we must have $\operatorname{Ker} D \subset \operatorname{Ker} A_\alpha$.

The multiplication of a vector by D^{-1} can be performed efficiently in the following way: The vector is extended with zero values in the nodes outside $\bar{\Omega}$. Then, the system of linear equations is solved with the matrix in (17) and the extended right-hand side. Since the triangulation $\mathcal{T}_\Pi$ is based on a rectangular grid, the matrix in (17) is separable, that is, after suitable permutation it can be expressed in a tensor product form. Therefore, the solution with this matrix can be performed with a fast direct solver, for example, the cyclic reduction method [40,42]. This requires $O(N \log N)$ floating point operations, where N is the dimension of the matrix B. Finally, the result of the multiplication by D^{-1} is the restriction of the solution of the extended problem to the nodes in $\bar{\Omega}$.

The next theorem guarantees that the condition number of the preconditioned system behaves well.

Theorem 4.1. *The discrete state problem corresponding to Neumann boundary conditions can be solved to a prescribed accuracy, independent of h, using $O(N \log N)$ floating point operations.*

Proof. From Assumption 3.1 and the results in [6], it follows that there exists a constant c_3, independent of the mesh step size h and the domain $\Omega_\alpha \in \mathcal{O}$, such that

$$\operatorname{Cond} D^{-1} A_\alpha \leq c_3$$

on the subspace $(\operatorname{Ker} A_\alpha)^\perp$. This implies that when the preconditioned conjugate gradient (PCG) method is used to compute the solution of (14) with a prescribed accuracy, the number of required iterations is bounded from above with a constant which is independent of h and Ω_α. As one iteration of the PCG method requires $O(N \log N)$ floating point operations the result follows.   □

4.2. Dirichlet boundary conditions

The direct discretization of the variational form (6) for the Dirichlet boundary value problem would lead to a system of linear equations for which it is not simple to construct an optimal preconditioner with respect to the condition number. Instead in our approach considered in [31, 39], the Dirichlet boundary conditions are enforced using boundary Lagrange multipliers; see [3, 8]. A related although different approach with boundary Lagrange multipliers is studied in [19]. For this purpose, we introduce the following bilinear forms and the following linear form:

$$a(u,v) = \int_\Omega (\nabla u \cdot \nabla v + c\,u\,v)\,\mathrm{d}x + \sigma \int_{\partial\Omega} u\,v\,\mathrm{d}s, \quad u,v \in H^1(\Omega), \tag{19}$$

$$m_b(\lambda,v) = \langle \lambda,\,v \rangle \qquad\qquad\qquad \lambda \in H^{-\frac{1}{2}}(\partial\Omega),\ v \in H^1(\Omega),$$

$$f(v) = \int_\Omega f\,v\,\mathrm{d}x, \qquad\qquad\qquad\qquad v \in H^1(\Omega),$$

where $\langle \cdot,\,\cdot \rangle$ denotes the duality pairing between $H^{-\frac{1}{2}}(\partial\Omega)$ and $H^{\frac{1}{2}}(\partial\Omega)$. The nonnegative constant σ in (19) makes the bilinear form $a(u,v)$ coercive. It is chosen such that $c + \sigma \geq \eta > 0$ with η independent of h. The coersivity of $a(\cdots,\cdots)$ is required for the construction of the block diagonal preconditioner in the form (22), (23); for details, see [28, 30, 31], for example. Now, we have the following saddle-point problem:

$$\begin{cases} \text{Find } (u,\lambda) \in H^1(\Omega) \times H^{-\frac{1}{2}}(\partial\Omega) \text{ such that} \\[2mm] a(u,v) + m_b(\lambda,v) = f(v), \\[1mm] m_b(\mu,u) = 0, \end{cases} \quad \forall(v,\mu) \in H^1(\Omega) \times H^{-\frac{1}{2}}(\partial\Omega). \tag{20}$$

Remark 4.2. For simplicity, we only consider the homogeneous Dirichlet boundary value problem. The non-homogeneous boundary condition $u = g$ on $\partial\Omega$, $g \in H^{\frac{1}{2}}(\partial\Omega)$, can be treated as follows: In the equations (20), the terms $\sigma \int_{\partial\Omega} g\,v\,\mathrm{d}s$ and $m_b(\mu,g)$ are added to the first and second right-hand side, respectively.

Similarly as with the Neumann boundary value problem, we define a continuous piecewise linear finite element subspace V_Ω approximating $H^1(\Omega)$ using triangulations $\mathcal{T}_\Omega$. For the Lagrange multipliers in $H^{-\frac{1}{2}}(\partial\Omega)$, we construct using V_Ω the finite element subspace

$$V_{\partial\Omega} = \mathrm{Trace}|_{\partial\Omega}\,V_\Omega.$$

This choice of $V_{\partial\Omega}$ implies that the duality pairings in the formulation (20) can be expressed by $L_2(\partial\Omega)$-inner products. By using the Courant basis functions for V_Ω and $V_{\partial\Omega}$, we obtain for (20) the system of linear equations

$$Cw = \begin{pmatrix} A & M_b^T \\ M_b & 0 \end{pmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \tag{21}$$

where the matrix blocks A and M_b correspond to the bilinear forms $a(u,v)$ and $m_b(\mu,u)$, respectively. Due to the choice of $V_{\partial\Omega}$, (21) has a unique solution which coincides in Ω with the standard finite element solution.

Now, the preconditioning matrix $\widehat{D}$ has the compatible block diagonal form

$$\widehat{D} = \begin{pmatrix} D & 0 \\ 0 & S \end{pmatrix}, \tag{22}$$

where the matrix block D is the same one as in (18). The matrix block S for the Lagrange multipliers has the form

$$S = M_{bb} \left(\Lambda^{\frac{1}{2}} + (c + \sigma)h_b I \right)^{-1} M_{bb}, \tag{23}$$

where M_{bb} is the $N_b \times N_b$ boundary mass matrix obtained as the restriction of the $N \times N_b$ matrix M_b on the boundary nodes. Here, h_b and N_b denote the characteristic mesh step size and the number of nodes on the boundary $\partial\Omega$. The matrix Λ is the scaled discrete finite element counterpart of the one-dimensional operator $-\frac{d^2}{ds^2}$ with the periodic boundary conditions. The mesh used to discretize this one-dimensional operator is uniform, and, thus, Λ has the form

$$\Lambda = \begin{pmatrix} 2 & -1 & & -1 \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ -1 & & -1 & 2 \end{pmatrix}. \tag{24}$$

The form (23), (24) for the preconditioner block S is obtained by applying the Friedrichs' inequality and by using a well-known Schur complement preconditioner for Poisson-type problems considered, for example, in [9, 11]. The multiplication of a vector by $\widehat{D}^{-1}$ can be computed efficiently as explained below. The multiplication by the matrix block D^{-1} can be performed in the same way as in Section 4.1. Since we use mass lumping, M_{bb} is a diagonal matrix and the multiplication by M_{bb}^{-1} is simple and inexpensive operation. The direct computation of the matrix $\Lambda^{\frac{1}{2}} + (c + \sigma)h_b I$ would be rather expensive, but the multiplication by it can be performed using the FFT requiring $O(N_b \log N_b)$ floating point operations. For details, see [9, 11], for example.

Remark 4.3. The quality of the preconditioner (23) deteriorates as $c + \sigma$ or the diameter of Ω grows. However, it works well under the assumptions that diameter of Ω is $O(1)$ and $0 < \eta \leq c + \sigma \leq \nu$ with η independent of h and ν is $O(1)$ [30].

Since the matrix C is symmetric but indefinite, the preconditioned conjugate gradient method cannot be used. For example, the preconditioned minimal residual (PMINRES) method is suitable for this kind of problems [20, 36].

Theorem 4.2. *The discrete state problem corresponding to Dirichlet boundary conditions can be solved to a prescribed accuracy, independent of h, using $O(N \log N)$ floating point operations.*

Proof. From Assumption 3.1, Theorem 4.1 and the results in [28, 39] it follows that the eigenvalues of the generalized eigenvalue problem

$$C w = \mu \widehat{D} w$$

belong to the set $[\eta_1, \eta_2] \cup [\eta_3, \eta_4]$, where the constants $\eta_1 \leq \eta_2 < 0 < \eta_3 \leq \eta_4$ are independent of the mesh step size h and the domain $\Omega_\alpha \in \mathcal{O}$. This guarantees that the number of required PMINRES iterations to reduce the norm of the residual by a prescribed factor is bounded from above with a constant which is independent of h and Ω_α. □

5. NUMERICAL EXPERIMENTS

5.1. Construction of the triangulation

Let us consider the geometric configuration shown in Figure 2. In order to construct triangulations, we parametrize the boundary curve $\partial\Omega$ by using two Bézier curves with each curve having nine control points,
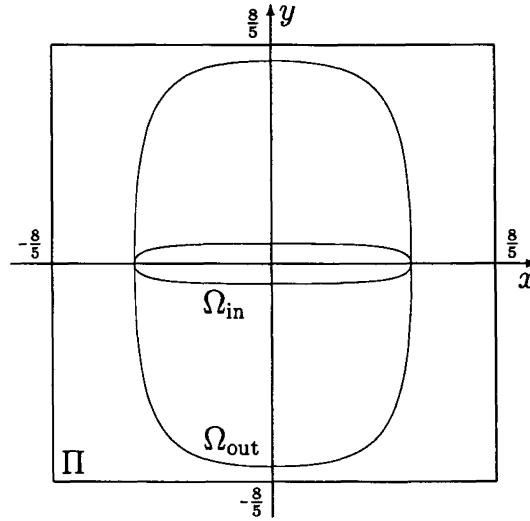
FIGURE 2. The geometrical configuration for our shape optimization problems.
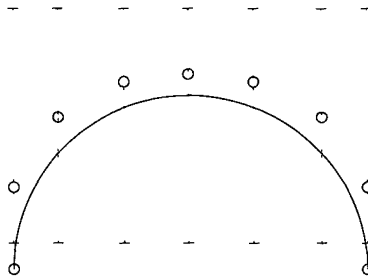

FIGURE 3. A Bézier curve approximation for a half of a circle, the Bézier control points and how they can move during the optimization.

one for the upper half and another one for the lower half. The control points on the points $(-1, 0)$ and $(1, 0)$ are fixed and the other control points are moving in the y-direction. The design variables α_k, $k = 1, \ldots, 14$, are the absolute values of the y-coordinates of the moving control points and we require that they satisfy the constraints $0.15 \leq \alpha_k \leq 1.5$. The set of admissible designs $\mathcal{O}$ contains all domains which are possible to obtain by using this parametrization and constraints. Thus, it holds that $\Omega \subset [-1, 1] \times [-\frac{3}{2}, \frac{3}{2}]$ for all $\Omega \subset \mathcal{O}$.

The initial domain Ω_0 for optimization is chosen to be the Bézier curve approximation of the circle of radius one and center at the origin. The Bézier curve approximation for the upper half of this circle is shown in Figure 3. The rectangle Π used in the construction of the triangulation for Ω is $\Pi = [-\frac{8}{5}, \frac{8}{5}] \times [-\frac{8}{5}, \frac{8}{5}]$.

The actual triangulation for an arbitrary domain $\Omega \subset \mathcal{O}$ is constructed in two steps. First, the triangulation $\mathcal{T}_{\Omega_0}$ is formed for the initial domain Ω_0 using the local fitting procedure introduced in [5]. An example of this is shown in Figure 1 (b). Also, the polygon P and the associated triangulation $\mathcal{T}_P$ are implicitly formed in this step. In the second step, the triangulation $\mathcal{T}_\Omega$ for Ω is obtained by moving the nodes of $\mathcal{T}_{\Omega_0}$. In this particular case, it is sufficient to move the nodes only in the y-direction. In the next paragraph, we explain how this is done in detail.

Let us consider a node (x^0, y^0) of the triangulation $\mathcal{T}_{\Omega_0}$ such that $y^0 \geq 0$. The case $y^0 \leq 0$ is treated analogously. Let the upper half of the boundary $\partial\Omega_0$ be defined by the Bézier curve $(b(t, \boldsymbol{u}^0), b(t, \boldsymbol{v}^0))$, where the polynomial $b(t, \boldsymbol{w}) = \sum_{i=0}^{k} \binom{k}{i} t^i (1 - t)^i w_i$ defines the components of the curve and the control points of

the curve are (u_i^0, v_i^0), $i = 0, \ldots, k$. There exists t such that $x^0 = b(t, \boldsymbol{u}^0)$ and $0 \leq t \leq 1$. Now, let $\boldsymbol{\alpha}$ be the vector containing the design variables defining the domain Ω and let the associated control points for the upper Bézier curve be given by $\boldsymbol{u}$ and $\boldsymbol{v}$. Then, only $\boldsymbol{v}$ depends on $\boldsymbol{\alpha}$ and $\boldsymbol{u} = \boldsymbol{u}^0$. The new coordinates (x, y) of the node (x^0, y^0) are given by $x = x^0$ and $y = b(t, \boldsymbol{v}) y^0 / b(t, \boldsymbol{v}^0)$.

The mesh for the initial domain fulfills the conditions in Assumption 3.1, since the local fitting procedure used satisfies the same conditions [5]. It is not so clear does this hold for the combined mesh generation procedure. When the Bézier curves defining the boundary are of sufficiently low order and the box constraints for the design parameters are strict enough, Assumption 3.1 holds. On the other hand when the Bézier curves are of sufficiently high order and the box constraints are loose enough, the boundary can oscillate so wildly that the mesh can be broken and, thus, Assumption 3.1 is not fulfilled. However, this situation is not desirable for any shape optimization algorithm and one usually tries to set the constraints in such a way that it can be avoided. It is difficult to give explicit limits for the degree of the Bézier curves and for the box constraints in order to Assumption 3.1 to hold. Hence, we will numerically study the validity of Assumption 3.1. In our shape optimization experiments, the numerical values for the constants c_1 and c_2 in Assumption 3.1 are given in Table 2 and Table 4.

5.2. Shape optimization with a Neumann BVP

We choose the state equation to be the Neumann boundary value problem:

$$
\begin{cases}
-\Delta u = f & \text{in } \Omega, \\
\dfrac{\partial u}{\partial n} = g & \text{on } \partial\Omega,
\end{cases}
\tag{25}
$$

where $g(x, y) = -2\sqrt{4 - 3x^2}$ and f is a function having a constant value which is chosen in such a way that the condition (4) is fulfilled.

The cost functional in the shape optimization problem measures the distance between the solution of the state equation (25) and a given target function u_d. The exact form of the cost functional is

$$
\mathcal{I}(\boldsymbol{\alpha}, u) = \frac{1}{2 \operatorname{Meas} \Omega} \int_\Omega (P(u - u_d))^2 \, \mathrm{d}x,
\tag{26}
$$

where the orthogonal projector $P : H^1(\Omega) \to \{v \in H^1(\Omega) \mid (v, 1)_{L_2(\Omega)} = 0\}$ is defined by

$$
P(v) = v - \frac{1}{\operatorname{Meas} \Omega} \int_\Omega v \, \mathrm{d}x.
$$

The purpose of the projector P in (26) is to make the value of $J(\boldsymbol{\alpha}, u)$ insensitive to functions which belong to the kernel of the operator defined by the state equation (25), that is, functions with a constant value. The target function in (26) is chosen to be

$$
u_d(x, y) = \frac{1}{2} - x^2 - 4y^2.
$$

The value of the cost functional for the ellipse $x^2 + 4y^2 = 1$ is zero and it is also an admissible design. Thus, the ellipse is a solution to this shape optimization problem.

In the forming of the system of linear equations (14), we are using mass lumping, that is, the numerical integrations are performed using the "trapezoid" quadrature rule. Then, for example, the right-hand side is given by

$$
\boldsymbol{f} = \boldsymbol{P}(\boldsymbol{M} \boldsymbol{f}_n + \boldsymbol{M}_b^T \boldsymbol{g}_n),
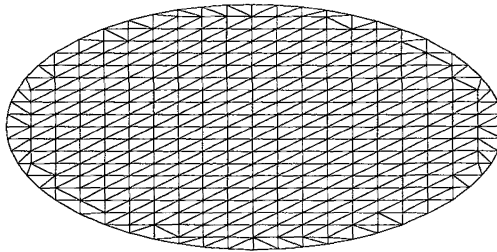\tag{27}
$$

FIGURE 4. The mesh for the target ellipse when $n = 33$.

TABLE 1. The results from the shape optimization with the Neumann bvp.

n	nodes	iter	eval	cost	dist	time
33	349	23	28	5.622×10^{-7}	2.424×10^{-2}	6.6
65	1313	28	29	2.810×10^{-8}	6.238×10^{-3}	27.7
129	5145	27	28	1.176×10^{-9}	1.233×10^{-3}	139.1
257	20321	23	27	5.894×10^{-11}	8.196×10^{-4}	561.6
513	80877	23	27	3.084×10^{-12}	7.710×10^{-4}	2750.4

where the vectors f_n and g_n contain the nodal values of $f(x, y)$ in Ω and $g(x, y)$ on $\partial\Omega$, respectively. Furthermore, M is the mass matrix and M_b is the matrix obtained by discretizing the bilinear form $m_b(\lambda, v)$ in (19). The matrix P in (27) is the orthogonal projection onto the subspace $\mathrm{Im}\, A$ and it is given by

$$P = I - \frac{1}{N} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix},$$

where N is the size of the problem. The discrete cost functional is

$$\mathcal{I}(\boldsymbol{\alpha}, \boldsymbol{u}) = \frac{1}{2\,\mathrm{Meas}\,\Omega}(\boldsymbol{u} - \boldsymbol{u}_d)^T P^T M P (\boldsymbol{u} - \boldsymbol{u}_d).$$

The optimization is performed using the E04UCF subroutine in the NAG library [35] which is an implementation of the SQP algorithm. In the iterative solution of the state and adjoint state equations, the initial guess for the solution is the zero vector and the stopping criterion is $\|r_k\|_{D^{-1}} \le 10^{-13}\|r_0\|_{D^{-1}}$. The vectors r_0 and r_k are the residuals computed from the initial guess and the approximation of the solution after k iterations, respectively.

In Figure 4, the target ellipse is shown when $n = 33$. If the mesh for the final design were plotted also into Figure 4, they would be less that one line width apart. We have collected the results from the optimization runs in Table 1. This table contains the following information: n is the number of nodes in both x- and y-direction in the rectangular mesh for Π. The number of nodes in Ω is "nodes". The number of SQP iterations and cost functional evaluations are "iter" and "eval", respectively. The final cost function value obtained in the optimization is "cost". The Euclidean distance between the vectors containing the design variables for the final design and the target design is "dist". The CPU seconds required to perform the optimization run on HP 9000/C160 workstation is given by "time".

Table 2 contains information about the linear problems and their solution during the optimization. This table contains the following information: n is the number of nodes in x- and y-direction in the mesh for Π. We have considered separately the solution of the state equation with the initial design, all solutions of the state

TABLE 2. The solution of linear problems during the shape optimization with the Neumann boundary value problem.

n	which	c_1	c_2	η_3	η_4	iter
33	first	3.703	1.516	0.322	1.301	19
33	all	15.336	2.528	0.154	2.497	34
33	last	13.920	2.371	0.162	2.388	34
65	first	3.986	1.755	0.327	1.230	20
65	all	13.593	2.977	0.156	2.423	34
65	last	12.097	2.742	0.165	2.288	37
129	first	5.843	1.721	0.329	1.557	25
129	all	18.326	2.427	0.157	2.637	36
129	last	16.785	2.215	0.165	2.481	39
257	first	4.942	1.600	0.330	1.285	22
257	all	16.742	2.772	0.157	2.569	36
257	last	15.229	2.530	0.166	2.412	36
513	first	5.511	1.794	0.331	1.394	24
513	all	17.998	3.180	0.157	2.822	38
513	last	15.957	2.829	0.166	2.647	40

equation and the adjoint state equations and the solution of the state equation with the final design. In "which" field, these are denoted by "first", "all" and "last", respectively. The upper bound for the condition numbers in Assumption 3.1 is c_1 and the upper bound for the ratios of the mesh step sizes on $\partial\Omega$ is c_2. The smallest nonzero eigenvalue and the largest eigenvalue of $\boldsymbol{D}^{-1}\boldsymbol{A}_\alpha$ are given by η_3 and η_4, respectively. The condition number in Theorem 4.1 is the ratio of these two values. The number of PCG iterations is "iter" and, in the case of all solutions, it gives the average number of iterations.

5.3. Shape optimization with a Dirichlet BVP

In this test example, the state equation is the Dirichlet boundary value problem:

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \tag{28}$$

where $f(x,y) = 10$. In this shape optimization problem, the cost functional is

$$J(\boldsymbol{\alpha}, u) = \frac{1}{2\,\text{Meas}\,\Omega} \int_\Omega (u - u_d)^2 \,\mathrm{d}x,$$

where the target function is $u_d(x,y) = 1 - x^2 - 4y^2$. Again, the ellipse $x^2 + 4y^2 = 1$ is a solution to this shape optimization problem.

This problem is discretized in the same way as the Neumann boundary value problem. In this case, the initial guess in the iterative solution of the state and adjoint state equations is the right-hand side vector multiplied by $\widehat{\boldsymbol{D}}^{-1}$. The stopping criterion for the iteration is the same one. The constant σ appearing in (19) and (23) has the value one. The approximate mesh step size on $\partial\Omega$ denoted by h_b in (23) has been chosen to be $1/N_b$, where N_b is the number of nodes on $\partial\Omega$.

Table 3 contains the same information as Table 1, but for the Dirichlet boundary value problem. Also, Table 4 is rather similar to Table 2. The eigenvalue bounds in Theorem 4.2 are given by η_1, η_2, η_3 and η_4 in Table 4. Now, "iter" is the number of PMINRES iterations. The SQP iteration histories are shown in Figure 5.

TABLE 3. The results from the shape optimization with the Dirichlet bvp.

n	nodes	iter	eval	cost	dist	time
33	349	18	28	2.235×10^{-7}	4.289×10^{-3}	10.9
65	1313	14	20	1.383×10^{-8}	2.768×10^{-3}	37.0
129	5145	17	27	8.628×10^{-10}	1.928×10^{-4}	250.1
257	20321	17	27	5.737×10^{-11}	2.479×10^{-4}	1123.1
513	80877	17	27	3.782×10^{-12}	7.140×10^{-5}	5805.7

TABLE 4. The solution of linear problems during the shape optimization with the Dirichlet boundary value problem.

| n	which	c_1	c_2	η_1	η_2	η_3	η_4	iter
33	first	3.703	1.516	-0.573	-0.420	0.830	1.408	31
33	all	145.785	3.436	-0.644	-0.265	0.187	7.728	66
33	last	13.950	2.295	-0.607	-0.351	0.508	2.448	61
65	first	3.986	1.755	-0.573	-0.409	0.772	1.337	34
65	all	124.860	4.163	-0.647	-0.221	0.159	7.522	76
65	last	12.101	2.715	-0.609	-0.337	0.502	2.335	65
129	first	5.843	1.721	-0.565	-0.403	0.735	1.614	37
129	all	149.764	3.474	-0.646	-0.217	0.153	8.178	78
129	last	16.784	2.212	-0.606	-0.327	0.501	2.523	69
257	first	4.942	1.600	-0.569	-0.386	0.728	1.373	38
257	all	131.741	3.932	-0.648	-0.197	0.150	7.986	82
257	last	15.228	2.530	-0.608	-0.297	0.481	2.447	71
513	first	5.511	1.794	-0.570	-0.381	0.725	1.467	41
513	all	162.768	4.390	-0.648	-0.196	0.149	8.784	88
513	last	15.957	2.829	-0.609	-0.290	0.469	2.678	76
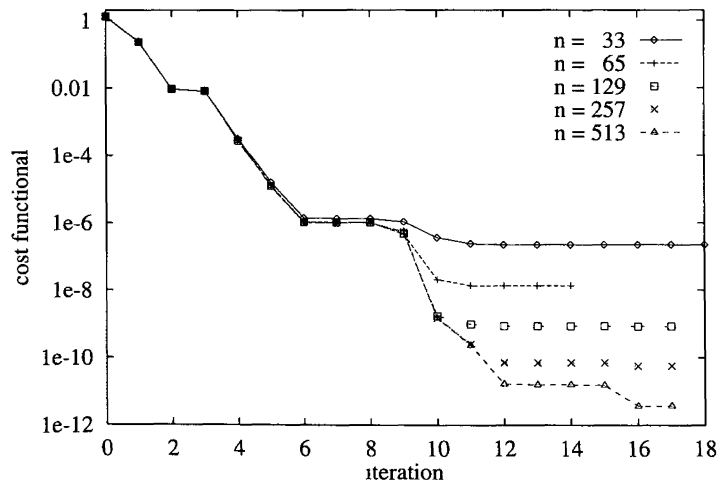
FIGURE 5. The SQP iteration histories for the shape optimization with the Dirichlet boundary value problem.

REFERENCES

[1] G.P. Astrakhantsev, Method of fictitious domains for a second–order elliptic equation with natural boundary conditions. *USSR Comput. Math. Math. Phys.* **18** (1978) 114–121.

[2] C. Atamian, G.V. Dinh, R. Glowinski, J. He and J. Périaux, On some imbedding methods applied to fluid dynamics and electro–magnetics. *Comput. Methods Appl. Mech. Engrg.* **91** (1991) 1271–1299.

[3] I. Babuška, The finite element method with Lagrangian multipliers. *Numer. Math.* **20** (1973) 179–192.

[4] A. Bespalov, Yu.A. Kuznetsov, O. Pironneau and M.-G. Vallet, Fictitious domain with separable preconditioners *versus* unstructured adapted meshes. *Impact Comput. Sci. Eng.* **4** (1992) 217–249.

[5] C. Börgers, A triangulation algorithm for fast elliptic solvers based on domain imbedding. *SIAM J. Numer. Anal.* **27** (1990) 1187–1196.

[6] C. Börgers and O.B. Widlund, On finite element domain imbedding methods. *SIAM J. Numer. Anal.* **27** (1990) 963–978.

[7] V. Braibant and C. Fleury, Shape optimal design using B–splines. *Comput. Methods Appl. Mech. Engrg.* **44** (1984) 247–267.

[8] J.H. Bramble, The Lagrangian multiplier method for Dirichlet's problem. *Math. Comp.* **37** (1981) 1–11.

[9] J.H. Bramble, J.E. Pasciak and A.H. Schatz, The construction of preconditioners for elliptic problems by substructuring, I. *Math. Comp.* **47** (1986) 103–134.

[10] R.A. Brockman, Geometric sensitivity analysis with isoparametric finite elements. *Comm. Appl. Numer. Math.* **3** (1987) 495–499.

[11] T.F. Chan, Analysis of preconditioners for domain decomposition. *SIAM J. Numer. Anal.* **24** (1987) 382–390.

[12] J. Daňková and J. Haslinger, Fictitious domain approach used in shape optimization: Neumann boudary condition, in Control of Partial Differential Equations and Applications (Laredo, 1994), *Lecture Notes in Pure and Appl. Math.*, Dekker, New York **174** (1996) 43–49.

[13] J. Daňková and J. Haslinger, Numerical realization of a fictitious domain approach used in shape optimization. I. Distributed controls. *Appl. Math.* **41** (1996) 123–147.

[14] P. Duysinx, W.H. Zhang and C. Fleury, Sensitivity analysis with unstructured free mesh generators in 2–D and 3–D shape optimization, in *Structural Optimization 93*, Vol. 2, Rio de Janeiro (1993) 205–212.

[15] P.E. Gill, W. Murray and M.H. Wright, Practical Optimization. Academic Press, New York (1981).

[16] R. Glowinski, T. Hesla, D.D. Joseph, T.-W. Pan and J. Périaux, Distributed Lagrange multiplier methods for particulate flows, in *Computational Science for the 21st Century*, M.-O. Bristeau, G. Etgen, W. Fitzgibbon, J.L. Lions, J. Périaux and M.F. Wheeler Eds., Wiley, Chichester (1997) 270–279.

[17] R. Glowinski and Yu.A. Kuznetsov, On the solution of the Dirichlet problem for linear elliptic operators by a distributed Lagrande multiplier method. *C.R. Acad. Sci. Paris Sér. I Math.* **327** (1998) 693–698.

[18] R. Glowinski, T.-W. Pan, A.J. Kearsley and J. Périaux, Numerical simulation and optimal shape for viscous flow by a fictitious domain method. *Internat. J. Numer. Methods Fluids* **20** (1995) 695–711.

[19] R. Glowinski, T.-W. Pan and J. Périaux, A fictitious domain method for Dirichlet problem and applications. *Comput. Methods Appl. Mech. Engrg.* **111** (1994) 283–303.

[20] A. Greenbaum, Iterative Methods for Solving Linear Systems. *Frontiers in Applied Mathematics,* SIAM, Philadelphia, PA, USA **17** (1997).

[21] J. Haslinger, Imbedding/control approach for solving optimal shape design problems. *East–West J. Numer. Math.* **1** (1993) 111–119.

[22] J. Haslinger, Comparison of different fictitious domain approaches used in shape optimization. Tech. Rep. 15, Laboratory of Scientific Computing, University of Jyväskylä (1996).

[23] J. Haslinger, K.H. Hoffmann and M. Kočvara, Control/fictitious domain method for solving optimal shape design problems. *RAIRO Modél. Math. Anal. Numér.* **27** (1993) 157–182.

[24] J. Haslinger and D. Jedelský, Genetic algorithms and fictitious domain based approaches in shape optimization. *Structural Optimization* **12** (1996) 257–264.

[25] J. Haslinger and A. Klarbring, Fictitious domain/mixed finite element approach for a class of optimal shape design problems. *RAIRO Modél. Math. Anal. Numér.* **29** (1995) 435–450.

[26] J. Haslinger and P. Neittaanmäki, *Finite Element Approximation for Optimal Shape, Material and Topology Design,* 2nd ed., Wiley, Chichester (1996).

[27] J. He, *Méthodes de domaines fictifs en méchanique des fluides applications aux écoulements potentiels instationnaires autour d'obstacles mobiles.* Ph.D. thesis, Université Paris VI (1994).

[28] E. Heikkola, Y. Kuznetsov, T. Rossi and P. Tarvainen, Efficient preconditioners based on fictitious domains for elliptic FE-problems with Lagrange multipliers, in ENUMATH 97 – Proceedings of the 2nd European Conference on Numerical Mathematics and Advanced Applications, H.G. Bock, G. Kanschat, R. Rannacher, F. Brezzi, R. Glowinski, Yu.A. Kuznetsov and J. Périaux Eds., World Scientific Publishing Co., Inc., River Edge, NJ (1998) 646–661.

[29] K. Kunisch and G. Peichl, Shape optimization for mixed boundary value problems based on an embedding method. *Dynam. Contin. Discrete Impuls. Systems* **4** (1998) 439–478.

[30] Yu.A. Kuznetsov, Efficient iterative solvers for elliptic finite element problems on nonmatching grids. *Russian J. Numer. Anal. Math. Modelling* **10** (1995) 187–211.

[31] Yu.A. Kuznetsov, Iterative analysis of finite element problems with Lagrange multipliers, in *Computational Science for the 21st Century*, M.-O. Bristeau, G. Etgen, W. Fitzgibbon, J.L. Lions, J. Périaux and M.F. Wheeler Eds., Wiley, Chichester (1997) 170–178.

[32] Yu.A. Kuznetsov and M.F. Wheeler, Optimal order substructuring preconditioners for mixed finite element methods on non-maching grids. *East–West J. Numer. Math.* **3** (1995) 127–143.

[33] R. Mäkinen, Finite–element design sensitivity analysis for non–linear potential problems. *Comm. Appl. Numer. Math.* **6** (1990) 343–350.

[34] G.I. Marchuk, Yu.A. Kuznetsov and A.M. Matsokin, Fictitious domain and domain decomposition methods. *Soviet J. Numer. Anal. Math. Modelling* **1** (1986) 3–35.

[35] NAG, *The NAG Fortran Library Manual: Mark 18.* NAG Ltd, Oxford (1997).

[36] C.C. Paige and M.A. Saunders, Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.* **12** (1975) 617–629.

[37] O. Pironneau, *Optimal Shape Design for Elliptic Systems.* Springer–Verlag, New York (1984).

[38] W. Proskurowski and P.S. Vassilevski, Preconditioning capacitance matrix problems in domain imbedding. *SIAM J. Sci. Comput.* **15** (1994) 77–88.

[39] T. Rossi, *Fictitious Domain Methods with Separable Preconditioners.* Ph.D. thesis, Department of Mathematics, University of Jyväskylä (1995).

[40] T. Rossi and J. Toivanen, A parallel fast direct solver for block tridiagonal systems with separable matrices of arbitrary dimension. *SIAM J. Sci. Comput.* **20** (1999) 1778–1793.

[41] J. Sokolowski and J.-P. Zolesio, *Introduction to Shape Optimization. Shape Sensitivity Analysis.* Springer–Verlag, Berlin (1992).

[42] P.N. Swarztrauber, The methods of cyclic reduction and Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle. *SIAM Rev.* **19** (1977) 490–501.

[43] J. Toivanen, *Fictitious Domain Method Applied to Shape Optimization.* Ph.D. thesis, Department of Mathematics, University of Jyväskylä (1997).

[44] L. Tomas, *Optimisation de Forme et Domaines Fictifs: Analyse de Nouvelles Formulations et Aspects Algorithmiques.* Ph.D. thesis, École Centrale de Lyon (1997).