2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum

Numerical design of an optimal bypass for a partially blocked artery

Rongliang Chen
College of Mathematics and Econometrics
Hunan University
Changsha, Hunan 410082, P. R. China
Email: rongliang.chen@colorado.edu

Xiao-Chuan Cai
Department of Computer Science
University of Colorado at Boulder
Boulder, CO 80309-0430, USA
Email: cai@cs.colorado.edu

***Abstract*—A parallel domain decomposition method is introduced for numerical design of an optimal bypass for a partially blocked artery. The optimal bypass is described as the solution of a shape optimization problem governed by the steady-state incompressible Navier-Stokes equations that are used to model the blood flow. The problem is discretized with a finite element method on unstructured moving meshes and then solved by a parallel one-shot Lagrange-Newton-Krylov-Schwarz algorithm. In order to accelerate the convergence of the inexact Newton method, we introduce a two-level inexact Newton method which solves a coarse grid problem to generate a good initial guess for the fine grid inexact Newton method. Numerical experiments show that our algorithms perform well on a supercomputer with hundreds of processors.**

***Keywords*-shape optimization; one-shot method; parallel computing; domain decomposition method; finite element method**

I. Introduction

Arterial stenosis is the narrowing of certain part of an artery and, in medical practice, the fix usually requires a new route (artery bypass graft) around the stenosed artery that allows the blood to flow smoothly in the artery*. There are some publications that study numerically the blood flow through arterial stenosis; see [13], [14], [15] and references therein. But there are few publications dedicated to the optimal design of the artery bypass graft; see e.g., [1], [19], [21] and [22]. The design problem is computationally very expensive, and large scale computing is absolutely necessary. In this paper, we use a parallel one-shot Lagrange-Newton-Krylov-Schwarz method, which has the potential to solve very large problems (e.g., three-dimensional problems on machines with a large number of processors) to numerically design the artery bypass graft of a partially blocked artery in two-dimensional domain.

The goal of the artery bypass design is to find the best shape of the bypass graft such that a certain property of the flow is optimized. In this paper we focus on the minimization of the shear stress (represented by the integral of the squared shear rate) of the blood flow which is a critically important criterion for artery bypass design problems [1]. Before introducing the mathematics model, we recall some notations: For a scalar function ϕ, a vector-valued function $\mathbf{u}=(u,v)$ and matrices $\mathbf{A}=(a_{ij})_{n\times n},\ \mathbf{B}=(b_{ij})_{n\times n}$, we denote

$$\nabla\phi := \left(\frac{\partial\phi}{\partial x},\ \frac{\partial\phi}{\partial y}\right),\quad \nabla\cdot\mathbf{u} := \frac{\partial u}{\partial x}+\frac{\partial v}{\partial y},$$

and

$$\mathbf{A}:\mathbf{B} := \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{ij}.$$

The artery bypass design problem is described mathematically by the following shape optimization problem governed by the steady-state incompressible Navier-Stokes equations defined in a two-dimensional domain Ω_α

$$\min_{\mathbf{u},\alpha}\quad J_o(\mathbf{u},\alpha) = 2\mu\int_{\Omega_\alpha}\varepsilon(\mathbf{u}):\varepsilon(\mathbf{u})dxdy + \frac{\beta}{2}\int_I(\alpha'')^2ds$$

subject to

$$\left\{\begin{array}{rcll} -\mu\Delta\mathbf{u}+\mathbf{u}\cdot\nabla\mathbf{u}+\nabla p &=& \mathbf{f} & \text{in } \Omega_\alpha,\\ \nabla\cdot\mathbf{u} &=& 0 & \text{in } \Omega_\alpha,\\ \mathbf{u} &=& \mathbf{g} & \text{on } \Gamma_{inlet},\\ \mathbf{u} &=& \mathbf{0} & \text{on } \Gamma_{wall},\\ \mu\dfrac{\partial\mathbf{u}}{\partial\mathbf{n}}-p\cdot\mathbf{n} &=& \mathbf{0} & \text{on } \Gamma_{outlet},\\ \alpha(a) &=& z_1, & \alpha(b) = z_2. \end{array}\right. \quad (1)$$

Here $\Omega_\alpha \in R^2$ is the computational domain and the subscript α is a shape function determined by a set of parameters $\mathbf{a}=(a_1,a_2,...,a_d)$ which determines the shape of the computational domain and d is the number of design variables. $\mathbf{u}=(u,v)$ and p represent the velocity and pressure of the blood flow, $\mathbf{n}$ is the outward unit normal vector on $\partial\Omega_\alpha$ and μ is the kinematic viscosity. Γ_{inlet}, Γ_{outlet} and Γ_{wall} represent the inlet, outlet and wall boundaries, respectively; see Figure 1. $\mathbf{f}$ is the given body force and $\mathbf{g}$ is the given velocity at the inlet Γ_{inlet}. In the constraints, the first five equations are the Navier-Stokes equations and boundary conditions that are used to model the blood flow and the last two equations indicate that the optimized boundary should be connected to the rest of the boundary and z_1 and z_2 are two given constants. In the objective function, $\varepsilon(\mathbf{u})=\frac{1}{2}(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)$ is the deformation tensor which reflects the shear stress of the blood flow [1] and

*www.reshealth.org/images/greystone/em_2405.gif

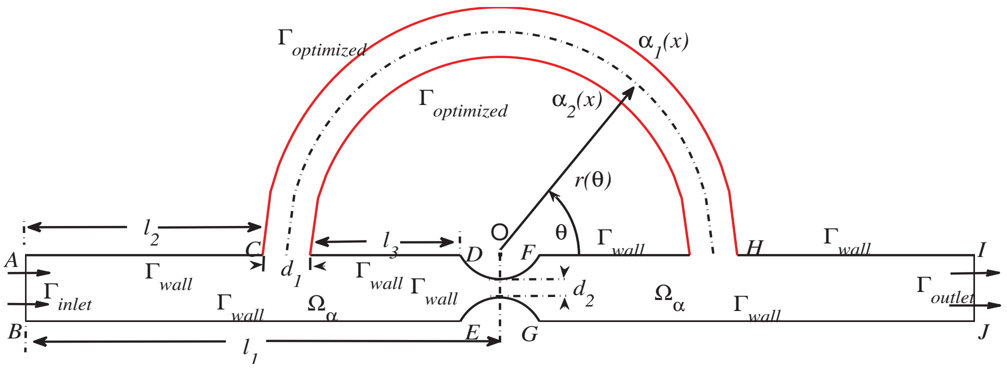

Figure 1. Simplified two-dimensional bypass model; the red boundary $\Gamma_{optimized}$ denotes the part of the boundary whose shape is to be determined by the optimization process. Here $l_1 = 6.0$, $l_2 = 3.0$, $l_3 = 1.9$, $d_1 = 0.6$, $d_2 = 0.2$ and $|AB| = 0.8$.

β is a nonnegative constant that controls the amount of regularization added to the objective function. $I = [a, b]$ is an interval in which the shape function $\alpha(x)$ is defined. The last term of the objective function is a regularization term providing the regularity of $\partial\Omega_\alpha$ [8]. In some approaches (e.g., [1] and [11]) some restrictions of the geometry are included in the constraints, e.g., certain thickness or volume, instead of a regularization term in the objective function.

II. Moving mesh finite element method for shape optimization problem

A. Mesh moving strategies

Since the computational domain of the shape optimization problem changes during the optimization process, the mesh needs to be modified to follow the computational domain. One can either reconstruct a new mesh, a process known as remeshing, or move the old mesh to obtain a new mesh following some rules, which is referred to as a moving mesh method. Remesh guarantees a good new mesh but it is computationally expensive and moving mesh changes the locations of the mesh points but keeps the number of mesh points and the connectivity unchanged, which is cheaper but the deformed mesh may become ill-conditioned when the boundary variation is large. In this paper, we focus on the moving mesh strategy where we only need to call the mesh generator once before the parallel solver begins. Another reason to use the moving mesh method instead of remesh is that we use the overlapping domain decomposition method in our algorithm and the moving mesh method does not change the mesh topology, so we can reuse the partition of the initial mesh when the mesh changes. In moving mesh methods, one can model the mesh movement either by a Laplace equation or a linear elasticity equation [25]. The elasticity equation is more difficult to solve than the Laplace equation but it often offers a better new mesh. We prefer the latter approach. Let α_0 be the initial shape of the boundary and Ω_{α_0} the initial computational domain. We define $\mathbf{x^0}$ as the coordinate of a point in Ω_{α_0}, $\mathbf{x}$ as the coordinate of the corresponding point in Ω_α and $\delta_\mathbf{x} := \mathbf{x} - \mathbf{x^0}$ as the displacement of this point. When the point $\mathbf{x^0}$ moves, we assume the displacement $\delta_\mathbf{x}$ satisfies the following equations:

$$\begin{cases} \nabla \cdot \sigma &= \mathbf{0} \quad \text{in} \quad \Omega_{\alpha_0}, \\ \delta_\mathbf{x} &= \mathbf{g}_\alpha \quad \text{on} \quad \partial\Omega_{\alpha_0}, \end{cases} \tag{2}$$

where σ is the stress tensor defined as

$$\sigma = \lambda \mathrm{Tr}(\epsilon)\mathbf{I} + 2\overline{\mu}\epsilon.$$

Here $\epsilon = \frac{1}{2}(\nabla\delta_\mathbf{x} + \nabla{\delta_\mathbf{x}}^\mathbf{T})$, Tr is the trace, $\mathbf{I}$ is an identity operator, λ and $\overline{\mu}$ are the Lamé constants. $\mathbf{g}_\alpha = (g^x_\alpha, g^y_\alpha)$ is the displacement on the boundary determined by the shape function $\alpha(x)$. Note that $\mathbf{g}_\alpha$ is not a given function, but a function obtained automatically during the iterative solution process.

B. Discretization of shape optimization problem

To numerically solve the shape optimization problem (1), one can either apply an optimization approach on the continuous level, e.g., the Lagrange multiplier method, to obtain the continuous optimal system and then discretize and solve it, or one can discretize problem (1) to obtain a finite dimensional constrained optimization problem and then use a discrete optimization method to solve it. These two strategies are called *optimize-then-discretize* (OTD) and *discretize-then-optimize* (DTO), respectively. The main difference of these two approaches is the boundary conditions for the adjoint equations associated with the Lagrange multipliers. In the OTD approach, the boundary conditions for the adjoint equations are obtained by the shape calculus [23] and in the DTO approach, such boundary conditions are not necessary since the adjoint problem is obtained algebraically. We only consider the DTO approach in this paper.

Our methods begin with discretizing the state equations (1) with a LBB-stable $Q_2 - Q_1$ finite element method and moving mesh equations (2) with a Q_2 finite element method to obtain a finite dimensional constrained optimization problem

$$\min_{\mathbf{u}^h,\mathbf{a},\delta_{\mathbf{x}}^h} \quad J_o^h(\mathbf{u}^h,\mathbf{a},\delta_{\mathbf{x}}^h) = \mu(\mathbf{u}^h)^T\mathbf{J}^h\mathbf{u}^h + \frac{\beta}{2}\mathbf{J}_\alpha$$

subject to

$$\left\{\begin{array}{rcl} \mathbf{K}^h\mathbf{u}^h + \mathbf{B}^h(\mathbf{u}^h)\mathbf{u}^h - \mathbf{Q}^h\mathbf{p}^h & = & \mathbf{F}_f^h + \mathbf{F}_u^h, \\ (\mathbf{Q}^h)^\mathbf{T}\mathbf{u}^h & = & \mathbf{0}, \\ \mathbf{D}^h\delta_{\mathbf{x}}^h & = & \mathbf{F}_x^h, \\ \mathbf{A}_\mathbf{a} & = & \mathbf{F}_\mathbf{a}, \end{array}\right. \tag{3}$$

where h is the mesh size parameter and $\mathbf{K}^h$, $\mathbf{B}^h(\mathbf{u}^h)$, $\mathbf{Q}^h$ and $\mathbf{D}^h$ are the coefficients of the discretized Navier-Stokes equations and linear elasticity equations. $\mathbf{F}_f^h$ is the discretized body force. $\mathbf{F}_u^h$ and $\mathbf{F}_x^h$ refer to the Dirichlet boundary conditions for $\mathbf{u}^h$ and $\delta_{\mathbf{x}}^h$, respectively, and $\mathbf{A}_\mathbf{a}$ and $\mathbf{F}_\mathbf{a}$ are the geometric constraints. $\mathbf{J}^h$ and $\mathbf{J}_\alpha$ are the coefficients of the discretized objective function. Note that $\mathbf{K}^h$, $\mathbf{B}^h(\mathbf{u}^h)$, $\mathbf{Q}^h$ and $\mathbf{J}^h$ depend on the grid displacement $\delta_{\mathbf{x}}^h$, while $\mathbf{D}^h$ is independent of $\delta_{\mathbf{x}}^h$. In our approaches, the mesh variable $\delta_{\mathbf{x}}^h$ is treated as an optimization variable and the moving mesh equations are viewed as constraints of the optimization problem which are solved simultaneously with the other equations. This makes our algorithms very simple, and it does not require a sensitivity analysis as in the traditional algorithms [24].

III. One-shot Lagrange-Newton-Krylov-Schwarz methods

In the last several decades, the typical approach for solving shape optimization problems considers the state variables $\mathbf{u}$ and p as functions of the design variable α and only regards the design variable as the optimization variable. The method is called nested analysis and design (NAND). NAND splits the optimality system of (1) into three components: the state equations for the constraints, the adjoint equations for the Lagrange multipliers, and the design equations for the shape design parameters, and then iteratively solves the three components; see [1], [16] and [21]. This method involves an iterative algorithm (nonlinear block Gauss-Seidel iteration) and needs to solve the state equations repeatedly, which is computationally expensive when the state equations are complicated, such as the Navier-Stokes equations. In addition, the three components have to be solved one after another; such a sequential approach is not desirable on machines with a large number of processors. An alternative to this approach is the *simultaneous analysis and design* (SAND), or the so-called one-shot approach, which views the state variables and design variable independently and solves the three components of the optimality system simultaneously (similar to applying a Newton-Krylov method to the fully coupled system); see [2], [9] and [10]. The challenges of the one-shot approach are that we need to solve a large nonlinear system (two to three times larger than the state equations), and the Jacobian system of the nonlinear system is also much larger and a lot more ill-conditioned. To answer these challenges, we need to design a good preconditioner that can substantially reduce the condition number of the large fully coupled system and, at the same time, provides the scalability for parallel computing. The advantages of one-shot approaches are their higher degree of parallelism and that none of the three components needs to be solved accurately until the end of the optimization process. In this paper, we study the one-shot method combined with the parallel Lagrange-Newton-Krylov-Schwarz (LNKSz) method for a particular shape optimization problem.

The one-shot LNKSz begins with reducing the discretized optimization problem (3) to a nonlinear equations problem (Karush-Kuhn-Tucker (KKT) system) using a Lagrange multiplier method [17]

$$\mathbf{G}^h(\mathbf{X}^h) \equiv \nabla_{\mathbf{X}^h}\mathbf{L}^h(\mathbf{X}^h) = \mathbf{0}, \tag{4}$$

where $\mathbf{X}^h \equiv \left(\mathbf{u}^h,\ \mathbf{p}^h,\ \delta_{\mathbf{x}}^h,\ \lambda_{\mathbf{u}}^h,\ \lambda_{\mathbf{p}}^h,\ \lambda_{\mathbf{x}}^h,\ \mathbf{a},\ \lambda^\mathbf{a}\right)^\mathbf{T}$ and $\mathbf{L}^h(\mathbf{X}^h)$ is the Lagrangian functional associated with problem (3) defined as

$$\begin{aligned} \mathbf{L}^h(\mathbf{X}^h) = {} & \mu(\mathbf{u}^h)^\mathbf{T}\mathbf{J}^h\mathbf{u}^h + \frac{\beta}{2}\mathbf{J}_\alpha \\ & +(\lambda_\mathbf{u}^h)^\mathbf{T}\cdot(\mathbf{K}^h\mathbf{u}^h + \mathbf{B}^h(\mathbf{u}^h)\mathbf{u}^h - \mathbf{Q}^h\mathbf{p}^h - \mathbf{F}_f^h - \mathbf{F}_u^h) \\ & +(\lambda_\mathbf{p}^h)^\mathbf{T}\cdot((\mathbf{Q}^h)^\mathbf{T}\mathbf{u}^h) \\ & +(\lambda_\mathbf{x}^h)^\mathbf{T}\cdot(\mathbf{D}^h\delta_\mathbf{x}^h - \mathbf{F}_x^h) + (\lambda^\mathbf{a})^\mathbf{T}\cdot(\mathbf{A}_\mathbf{a} - \mathbf{F}_\mathbf{a}), \end{aligned}$$

and $\lambda_\mathbf{u}^h$, $\lambda_\mathbf{p}^h$, $\lambda_\mathbf{x}^h$ and $\lambda^\mathbf{a}$ are the Lagrange multipliers for the equality constraints.

Then the nonlinear KKT system (4) is solved by an inexact Newton method which begins with a given initial guess $\mathbf{X}_0^h$, at each iteration, $k = 0, 1, \cdots$, a search direction $\mathbf{d}_k^h$ is obtained by approximately solve the right-preconditioned system satisfying

$$\| \mathbf{H}_k^h(\mathbf{M}_k^h)^{-1}(\mathbf{M}_k^h\mathbf{d}_k^h) + \mathbf{G}_k^h \| \le \max\{\eta_r^h \| \mathbf{G}_k^h \|, \eta_a^h\}, \tag{5}$$

where $\mathbf{H}_k^h = \nabla_{\mathbf{X}^h}\mathbf{G}^h(\mathbf{X}_k^h)$ is the Jacobian matrix of the nonlinear system (4), $\mathbf{G}_k^h = \mathbf{G}^h(\mathbf{X}_k^h)$ and η_r^h and η_a^h are relative and absolute tolerances for the linear solver. $(\mathbf{M}_k^h)^{-1}$ is an additive Schwarz preconditioner to be defined below.

To define the additive Schwarz preconditioner, we need an overlapping partition of Ω_α. Since the mesh topology doesn't change when the shape of the domain changes, we obtain the partition using the initial mesh on Ω_{α_0}. We first partition the domain Ω_{α_0} into non-overlapping subdomains Ω_{α_l}, $l = 1, \cdots, N_p$ and then extend each subdomain Ω_{α_l} to $\Omega_{\alpha_l}^\delta$ which overlaps its neighbors, i.e., $\Omega_{\alpha_l} \subset \Omega_{\alpha_l}^\delta$. Here δ is the size of the overlap which is understood in terms of the number of elements; i.e., $\delta = 8$ means the overlapping size is 8 layers of elements, and N_p is the number of subdomains which is equal to the number of processors (np) in this paper.

Then the additive Schwarz preconditioner is defined as [26]

$$(\mathbf{M}_k^h)^{-1} = \sum_{l=1}^{N_p} (R_l^\delta)^{\mathbf{T}} (\mathbf{H}_k^h)_l^{-1} R_l^\delta,$$

where $(\mathbf{H}_k^h)_l = R_l^\delta \ \mathbf{H}_k^h \ (R_l^\delta)^{\mathbf{T}}$, R_l^δ is a restriction operator from Ω_α to the l^{th} overlapping subdomain. The subdomain preconditioners $(\mathbf{H}_k^h)_l^{-1} (l = 1, 2, \cdots, N_p)$ are computed by a sparse LU factorization. After approximately solving (5), we use a line search approach to globalize the inexact Newton method, where the new approximate solution is

$$\mathbf{X}^{k+1} = \mathbf{X}^k + \tau^k \mathbf{d}^k,$$

and the step length τ^k is selected by a cubic line search method [6].

Finding a good initial guess is very important for Newton type methods. In this paper, we try to find a good initial guess for problem (4) by solving a coarse problem which is defined with exactly the same method as the original discrete optimization problem but on a coarser grid, i.e., the initial guess $\mathbf{X}_0^h = I_H^h \mathbf{X}^H$, where I_H^h is a coarse to fine interpolation operator to be defined shortly and $\mathbf{X}^H$ is the solution of a coarse grid problem denoted as

$$\mathbf{G}^H(\mathbf{X}^H) = \mathbf{0}, \tag{6}$$

i.e., the KKT system on a coarse grid. The coarse-level problem (6) is solved by a similar inexact Newton method (change the subscript h to H) as that for the fine-level problem described above. We call this method a two-level inexact Newton method ([5], [27]).

The coarse to fine interpolation operator I_H^h is a $N_f \times N_c$ matrix, where N_f and N_c are the degrees of freedom on the fine grid and the coarse grid, respectively. The components of I_H^h are defined as

$$(I_H^h)_{ij} = \phi_j^H(\mathbf{x}_i^h), \tag{7}$$

i.e. the value of the j^{th} coarse grid function ϕ_j^H at the i^{th} fine grid point $\mathbf{x}_i^h$. In practice the function $\phi_j^H(\mathbf{x})$ doesn't have to be the same as the finite element basis function used to generate the coarse-level problem, in fact, we use here a radial basis function [5]. We tested some other interpolation methods and concluded that the radial basis function is more efficient for our problem. Unfortunately, the initial guess $\mathbf{X}_0^h$ interpolated from the coarse-level solution $\mathbf{X}^H$ using, straightforwardly, the interpolation matrix (7) does not reduce the initial residual of the fine-level problem as expected. This is because the Lagrange multiplier $\lambda_{\mathbf{x}}^h$ has a sharp jump at the moving boundary which the interpolation operator (7) cannot resolve on the coarse grid. To resolve this jump, we have to modify the interpolation operator in a way that is motivated by the "pollution removing technique" introduced in [20]. Besides the sharp jump at the moving boundary, another observation is that the value of $\lambda_{\mathbf{x}}^h$ on the moving boundary decreases following the mesh refinement. To illustrate this situation, we take the derivative of the Lagrangian functional $\mathbf{L}^h(\mathbf{X}^h)$ with respect to the design variable α

$$\sum_{i=1}^{n_x} \left(\overline{\lambda}_i^x \frac{\partial \mathbf{g}_\alpha^x(\mathbf{x}_i^h)}{\partial \alpha} + \overline{\lambda}_i^y \frac{\partial \mathbf{g}_\alpha^y(\mathbf{x}_i^h)}{\partial \alpha} \right) = -\frac{\beta}{2} \frac{\partial \mathbf{J}_\alpha}{\partial \alpha}, \tag{8}$$

where $\overline{\lambda}_i^x$ and $\overline{\lambda}_i^y$ $(i = 1, 2, \cdots, n_x)$ are the Lagrange multipliers related to the mesh displacements $\delta_{x_i}^h$ and $\delta_{y_i}^h$ $(i = 1, 2, \cdots, n_x)$ on the moving boundary, n_x is the number of finite element nodes on the moving boundary and $\mathbf{x}_i^h$ is the coordinate of the i^{th} node on the moving boundary. In equation (8), $\partial \mathbf{J}_\alpha / \partial \alpha$, $\partial \mathbf{g}_\alpha^x / \partial \alpha$ and $\partial \mathbf{g}_\alpha^y / \partial \alpha$ are functions of α that change a little following the mesh refinement but n_x increases a lot following the same mesh refinement, as a result, the values of $\overline{\lambda}_i^x$ and $\overline{\lambda}_i^y$ $(i = 1, 2, \cdots, n_x)$ decrease following the mesh refinement. The standard interpolation operator (7) does not respect this phenomenon. In order to reflect this, we modify the components of the interpolation operator (7) related to the Lagrange multipliers $\overline{\lambda}_i^x$ and $\overline{\lambda}_i^y$ $(i = 1, 2, \cdots, n_x)$ on the moving boundary by dividing a factor $\gamma = n_x^h / n_x^H$, where n_x^h and n_x^H are the numbers of finite element nodes on the moving boundary of the fine and coarse meshes, respectively. We show numerically in the next section that the interpolation operator (7) combined with the modifications works very well for the problem under consideration.

IV. Numerical experiments

In this section, we present some numerical results concerning the optimal design of the artery bypass graft and we mainly focus on the parallel performance of the domain decomposition preconditioner which is the most critical component of the one-shot approach. Our algorithm is implemented using PETSc [3], a Portable, Extensible Toolkit for Scientific computation developed at Argonne National Laboratory. All computations are performed on a Dell PowerEdge C6100 supercomputer (1368 compute nodes, each node contains two hex-core 2.8Ghz Intel Westmere processors and the nodes are interconnected via a non-blocking QDR Infiniband high performance network theoretically capable of 40Gbps) at the University of Colorado at Boulder. Unstructured meshes generated with CUBIT [18] from Sandia National Laboratory and partitioned with ParMETIS [12] are used in the numerical experiments.

Without the blockage (as in Figure 1) the blood flow is supposed to go from AB to IJ, but now we assume that the artery is partly blocked at DE and we need a bypass CH to let the blood go through. For simplicity, the thickness of CH is fixed. The goal is to find the best shape of the bypass CH, such that the shear stress of the fluid in the entire computational domain Ω_α is minimized. For simplicity, we let the body forces $\mathbf{f} = \mathbf{0}$ in the state equations (1). The boundary condition on the inlet Γ_{inlet} is chosen as a constant v_{in}, no-slip boundary conditions are used on the walls Γ_{wall} and on the outlet boundary Γ_{outlet} the stress-free boundary conditions are imposed; see (1). We use a polynomial function

$$r(\theta) = \sum_{i=1}^{d} r_i \theta^i,$$

with $d = 5$ to represent the centerline of the bypass (see the dashed line in Figure 1) whose shape is to be determined by the optimization process. Other shape functions can be used, but here we simply follow the paper [1]. In the moving mesh equations (2), the Lamé constants λ and $\overline{\mu}$ are are related to the Young's modulus E and the Poisson's ratio ν by

$$\lambda = \frac{\nu E}{(1+\nu)(1+\nu)}, \quad \overline{\mu} = \frac{E}{2(1+\nu)}.$$

$E = 10000$ and $\nu = 0$ are used in our experiments as suggested in [4].

An analytic Jacobian matrix is used in all the experiments. The Jacobian system in each Newton step is solved by a right-preconditioned GMRES with an absolute tolerance of 10^{-10} and a relative tolerance of 10^{-3}. The Newton iteration is stopped when the nonlinear residual is decreased by a factor of 10^{-6}.

Figure 2 shows the shear rate distribution of the initial and the optimal shape of the bypass graft where the shear stress of the optimal shape (bottom) is 10.9% less than that of the initial shape (top). The centerline of the initial shape is a semicircle of radius 2.7 and the centerline of the computed optimal shape is

$$r(\theta) = -0.005\theta^5 + 0.042\theta^4 - 0.1\theta^3 + 0.342\theta^2 - 0.866\theta + 2.7.$$

Figure 3 and Figure 4 show the streamline and pressure distribution of the initial shape and the optimal shape. In Figure 3, we can see that there is more vorticity in the initial shape than that in the optimal shape and Figure 4 shows that the maximum pressure of the initial shape (35.81) is larger than that of the optimal shape (32.47).

The parallel performance of the one-level and two-level inexact Newton methods is shown in Table I, where the number of Newton iterations (Newton) stays as a constant when the number of processors increases and the average number of GMRES iterations per Newton step (GMRES) increases slowly following the increase of np. From Table I, we can see that the two-level inexact Newton method not only reduces the number of Newton iterations, but also reduces the average number of GMRES iterations in each Newton step. The headings "Time" and "Time ratio (coarse/fine)" in Table I refer to the total compute time in seconds and the percentage of the total compute time spent on solving the coarse-level problem, respectively. The speedup curves of the one-level and two-level inexact Newton methods are shown in Figure 5, where the blue line is the linear speedup which means that doubling the number of processors halves

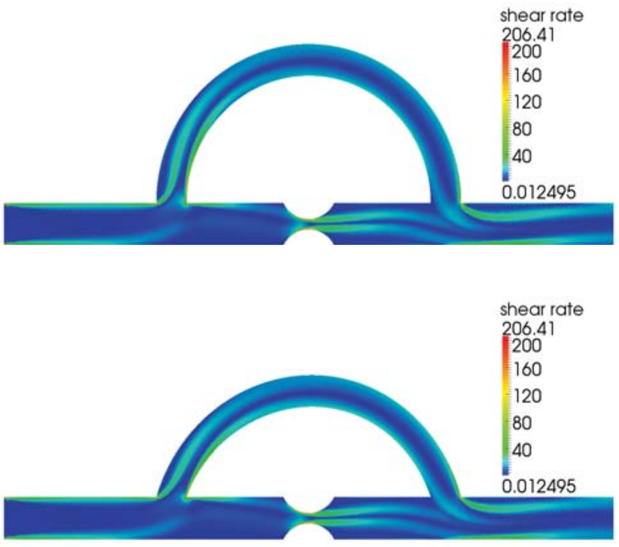

Figure 2. Shear rate distribution of the initial shape (top) and the optimal shape (bottom). Here $\beta = 5.0$ and $Re = 300$.

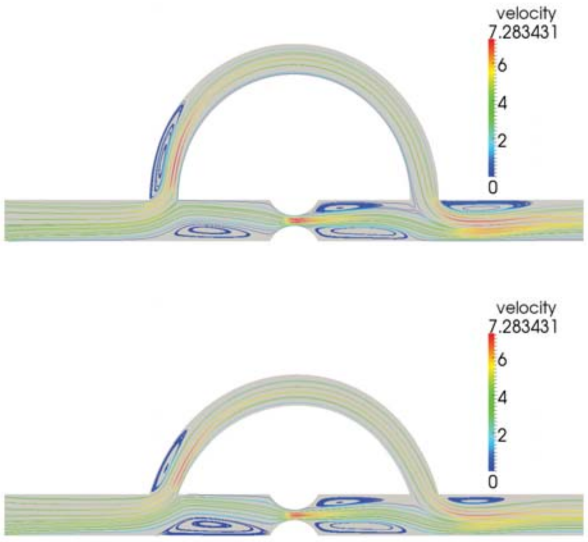

Figure 3. The streamline of the initial shape (top) and the optimal shape (bottom). Here $\beta = 5.0$ and $Re = 300$.

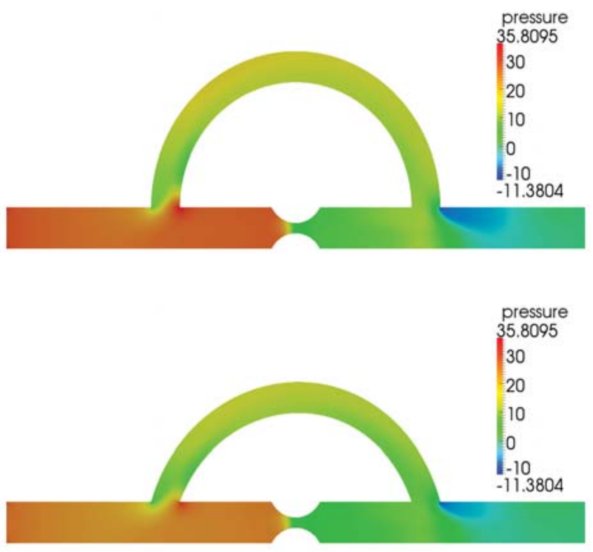

Figure 4. Pressure distribution of the initial shape (top) and the optimal shape (bottom). Here $\beta = 5.0$ and $Re = 300$.

Table I
Comparison of the one-level inexact Newton method (One) and the two-level inexact Newton method (Two). Here the degrees of freedom on the fine grid is 1.6×10^6 and that on the coarse grid is 4.0×10^5.

np	Newton		GMRES		Time		Time ratio (coarse/fine)	
	One	Two	One	Two	One	Two	One	Two
32	6	5	162.83	136.00	2393.08	2229.84	0	6.8%
64	6	5	191.00	169.40	1004.60	864.04	0	12.7%
128	6	5	254.83	203.80	681.68	552.43	0	33.3%
256	6	5	303.67	209.60	526.66	380.06	0	43.5%

the total compute time, and the red line and pink line refer to the speedup of the two-level method and one-level method, respectively. It is clear that the two-level method is better when the number of processors is large.

In overlapping domain decomposition methods, the overlap size δ is a very important parameter for adjusting the strength of the preconditioner. Increasing δ can reduce the number of GMRES iterations but increases the amount of information transfer between subdomains which is expensive. Table II shows the performance of the algorithms with different values of the overlap size δ. To obtain good results in terms of the total compute time, the overlap has to be quite large. This is very different from solving scalar elliptic equations where small overlap ($\delta = 1$) is usually enough [7]. Table III shows the performance of our method with respect to various Reynolds numbers Re and regularization parameter β. The heading "Reduction" in Table III refers to the reduction of the shear stress. A higher percentage of

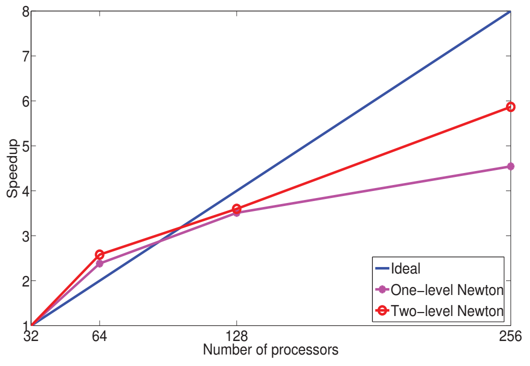

Figure 5. The speedup of the one-level and two-level inexact Newton methods for two different mesh sizes (the speedup is relatively measured to the performance obtained with 32 processors).

Table II
The effect of the overlap size δ for the one-level inexact Newton method and the inexact two-level Newton method. Here the degrees of freedom is 1.6×10^6 on the fine grid and 4.0×10^5 on the coarse grid .

np	δ	Newton		GMRES		Time	
		One	Two	One	Two	One	Two
64	14	6	7	228.50	181.14	1016.32	1025.11
64	16	6	5	191.00	169.00	1005.59	868.80
64	18	6	5	172.50	157.80	1131.80	897.23
128	14	6	5	271.67	231.00	707.86	525.30
128	16	6	5	254.33	203.80	674.83	552.43
128	18	6	5	209.33	196.67	732.49	619.67

reduction is achieved when the Reynolds number is large or the regularization parameter is small but the problem becomes harder to solve when the Reynolds number is increased or the regularization parameter is decreased.

V. Conclusion

In this paper, we developed a parallel one-shot Lagrange-Newton-Krylov-Schwarz method for an artery bypass design problem. We obtained an optimal bypass graft which reduces the shear stress by over 10%. In order to accelerate the convergence of the inexact Newton method, we introduced a two-level inexact Newton method which solves a coarse grid problem to generate a good initial guess for the fine grid inexact Newton method. The numerical experiments show that our algorithm is very efficient for large problem (over 2.8 million degrees of freedom) on a machine with up to 256 processors. The two-level inexact Newton method is robust with respect to the Reynolds number Re and the regularization parameter β, and its strong scalability is also better than the one-level inexact Newton method. Even though we only studied an artery bypass design problem in this paper, our algorithm can be extended to other shape optimization problems and three-dimensional problems.

Table III
The effect of the Reynolds number (Re) and the regularization parameter (β) for the one-level inexact Newton method and the two-level inexact Newton method. Here the degrees of freedom is 2.8×10^6.

Re	β	Newton		GMRES		Time		Time ratio (coarse/fine)		Reduction
		One	Two	One	Two	One	Two	One	Two	
100	15	5	5	122.80	162.20	918.79	957.48	0	4.9%	0.3%
200	15	4	5	223.00	210.60	921.62	1016.76	0	7.8%	2.4%
300	15	7	5	266.71	230.20	1538.11	1071.04	0	25.8%	6.7%
100	12	5	5	122.80	167.80	901.38	960.48	0	4.7%	0.4%
200	12	4	5	233.00	213.00	933.93	1043.07	0	7.6%	2.9%
300	12	10	5	292.40	230.20	2157.92	1072.93	0	27.3%	7.7%
100	10	6	5	127.83	168.80	1024.61	1026.46	0	4.6%	0.5%
200	10	5	5	249.20	215.40	1125.70	1067.06	0	7.7%	3.3%
300	10	20	5	327.50	219.00	4378.57	1039.10	0	33.7%	8.5%

Acknowledgment

We would like to thank Andrew T. Barker, Yuqi Wu, and Chao Yang for many discussions and comments and the PETSc team of Argonne National Laboratory for their help on using the PETSc library. We would like to acknowledge the support of NSF under DMS-0913089. The work of the first author was done while visiting the University of Colorado at Boulder and was partially supported by China Scholarship Council (CSC) and NSF China under No. 10971058.

References

[1] F. Abraham, M. Behr, and M. Heinkenschloss, *Shape optimization in stationary blood flow: A numerical study of non-Newtonian effects*, Comput. Methods Biomech. Biomed. Engrg., 8 (2005), pp. 127-137.

[2] E. Arian and S. Ta'asan, *Shape optimization in one-shot*, In: Optimal Design and Control, J. Boggaard et al (Eds.), Birkhauser, Boston, 1995, pp. 273-294.

[3] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, *PETSc Users Manual*, Technical report ANL-95/11, Argonne National Laboratory, 2011.

[4] R. T. Biedron and J. L. Thomas, *Recent enhancements to the FUN3D flow solver for moving-mesh applications*, AIAA-2009-1360, 2009.

[5] A. Barker and X.-C. Cai, *Two-level Newton and hybrid Schwarz preconditioners for fluid-structure interaction*, SIAM J. Sci. Comput., 32 (2010), pp. 2395-2417.

[6] J. E. Dennis and R. B. Schnable, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.

[7] M. Dryja and O. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comput., 15 (1994), pp. 604-620.

[8] M. D. Gunzburger, *Perspectives in Flow Control and Optimization: Advances in Design and Control*, SIAM, Philadelphia, 2003.

[9] O. Ghattas and C. Orozco, *A parallel reduced Hessian SQP method for shape optimization*, In: Natalia M. Alexandrov and M.Y. Hussaini, eds., Multidisciplinary Design Optimization: State of the Art, SIAM, Philadelphia, 1997, pp. 133-152.

[10] S. B. Hazra, *Multigrid one-shot method for aerodynamic shape optimization*, SIAM J. Sci. Comput., 30 (2008), pp. 1527-1547.

[11] B. He, O. Ghattas, and J. F. Antaki, *Computational strategies for shape optimization of time dependent Navier-Stokes flow*, Technical Report CMU-CML-97-102, Carnegie-Mellon University, 1997.

[12] G. Karypis, *METIS/ParMETIS web page*, University of Minnesota, 2011. http://glaros.dtc.umn.edu/gkhome/views/metis.

[13] M. Li, J. Beech-Brandt, L. John, P. Hoskins, and W. Easson, *Numerical analysis of pulsatile blood flow and vessel wall mechanics in different degrees of stenoses*, J. Biomech., 40 (2007), pp. 3715-3724.

[14] Q. Long, L. Luppi, C. Konig, V. Rinaldo, and S. Das, *Study of the collateral capacity of the circle of Willis of patients with severe carotid artery stenosis by 3D computational modeling*, J. Biomech., 41 (2008), pp. 2735-2742.

[15] Q. Long, X. Xu, K. Ramnarine, and P. Hoskins, *Numerical investigation of physiologically realistic pulsatile flow through arterial stenosis*, J. Biomech., 34 (2001), pp. 1229-1241.

[16] B. Mohammadi and O. Pironneau, *Optimal shape design for fluids*, Annu. Rev. Fluid Mech., 36 (2004), pp. 255-279.

[17] J. Nocedal and S. J. Wright, *Numerical Optimization*, First ed., Springer-Verlag, Berlin, 2000.

[18] S. J. Owen and J. F. Shepherd, *CUBIT project web page*, 2011, http://cubit.sandia.gov/.

[19] M. Probst, M. Lülfesmann, M. Nicolai, H. Bücker, M. Behr, and C. Bischof, *Sensitivity of optimal shapes of artificial grafts with respect to flow parameters*, Comput. Methods Appl. Mech. Engrg., 199 (2010), pp. 997-1005.

[20] E. Prudencio, R. Byrd, and X.-C. Cai, *Parallel multilevel restricted Schwarz preconditioners with pollution removing for PDE-constrained optimization*, SIAM J. Sci. Comput., 29 (2007), pp. 964-985.

[21] A. Quarteroni and G. Rozza, *Optimal control and shape optimization of aorto-coronaric bypass anastomoses*, Math. Models and Methods in Appl. Sci., 13 (2003), pp. 1801-1823.

[22] S. Sankaran and A. L. Marsden, *The impact of uncertainty on shape optimization of idealized bypass graft models in unsteady flow*, Phys. Fluids, 22 (2010), 121902.

[23] S. Schmidt and V. Schulz, *Shape derivatives for general objective functions and the incompressible Navier-Stokes equations*, Control Cybern., 39 (2010), pp. 677-713.

[24] J. Sokolowski and J.-P. Zolesio, *Introduction to Shape Optimization: Shape Sensitivity Analysis*, Springer-Verlag, Berlin, 1992.

[25] A. H. Truong, C. A. Oldfield, and D. W. Zingg, *Mesh movement for a discrete-adjoint Newton-Krylov algorithm for aerodynamic optimization*, AIAA J., 46 (2008), pp. 1695-1704.

[26] A. Toselli and O. Widlund, *Domain Decomposition Methods: Algorithms and Theory*, Springer-Verlag, Berlin, 2005.

[27] Y. Wu and X.-C. Cai, *A parallel two-level method for simulating blood flows in branching arteries with the resistive boundary condition*, Comput. & Fluids 45 (2011) 92-102.