Computers & Fluids xxx (2011) xxx–xxx

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

A parallel two-level method for simulating blood flows in branching arteries with the resistive boundary condition ☆

Yuqi Wu [a], Xiao-Chuan Cai [b,*]

[a] Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO 80309, United States
[b] Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309, United States

ARTICLE INFO

Article history:
Received 5 September 2010
Received in revised form 14 November 2010
Accepted 17 November 2010
Available online xxxx

Keywords:
Fluid–structure interaction
Blood flow models
Resistive boundary condition
Arbitrary Lagrangian–Eulerian framework
Multilevel domain decomposition
Preconditioning
Parallel computing

ABSTRACT

Computer modeling of blood flows in the arteries is an important and very challenging problem. In order to understand, computationally, the sophisticated hemodynamics in the arteries, it is essential to couple the fluid flow and the elastic wall structure effectively and specify physiologically realistic boundary conditions. The computation is expensive and the parallel scalability of the solution algorithm is a key issue of the simulation. In this paper, we introduce and study a parallel two-level Newton–Krylov–Schwarz method for simulating blood flows in compliant branching arteries by using a fully coupled system of linear elasticity equation and incompressible Navier–Stokes equations with the resistive boundary condition. We first focus on the accuracy of the resistive boundary condition by comparing it with the standard pressure type boundary condition. We then show the parallel scalability results of the two-level approach obtained on a supercomputer with a large number of processors and on problems with millions of unknowns.

1. Introduction

Computer modeling is a useful tool for the study of hemodynamics of blood flows in human arteries. Accurate modeling is important in prediction and treatment of, for example, cardiovascular diseases [8,20,23]. The extensive computational cost of the simulation demands scalable algorithms and large scale supercomputers. In this paper, we develop and study a parallel fluid–structure interaction (FSI) algorithm for the simulation of blood flows in branching arteries by using a fully coupled system of partial differential equations. Effective coupling of the fluid and the wall deformation is one of the main concerns in FSI. We adopt a monolithic approach by coupling the fluid, the elastic wall and fluid mesh movement equations in the arbitrary Lagrangian–Eulerian (ALE) framework [26], where the coupling conditions are enforced strongly as part of the system [3–6,16]. In [3,4], a class of parallel scalable Newton–Krylov–Schwarz (NKS) method was introduced for the FSI problem by using the one- and two-level Schwarz algorithms. Only the zero-traction outflow boundary condition was investigated in the papers. In this paper we extend the work of [3,4] to the case with a more physiologically realistic resistance outflow boundary condition. Since this boundary condition is of integral type, it changes not only the physics of the flow, but also the behavior of the preconditioning algorithms.

The vascular system is large and it is difficult to simulate the entire system due to the lack of computing power. In most of the simulations, numerical models are truncated into the upstream domain (domain of interest) and downstream domain, and an appropriate boundary condition is specified to represent the contributions of the downstream vascular network. The accuracy of the simulation depends on the outflow boundary condition which is imposed to represent the influence from absent parts of the circulation system. If the zero-pressure or zero-traction boundary condition is used on the outlets, it has been reported that the blood pressure is not computed accurately, and in the case of branching arteries with outlets of different sizes, the flow distribution is computed inconsistently with the clinical measurements [13,25]. Recent studies [14,18] suggest to use a reduced dimensional model to represent the downstream vessels and provide boundary condition for the domain of interest. In our application, the resistance outflow boundary condition is considered, where we assume the pressure P are constants over the upstream outlets. The relation $P = QR$ is implicitly prescribed on the outflow boundaries as the resistance boundary condition, where $Q = \int_{\Gamma_o} \mathbf{u} \cdot \mathbf{n} ds$ represents the flow rate at the outflow boundary and R is the measured resistance [24,25].

☆ This research was supported in part by DOE under DE-FC-02-06ER26784, and in part by NSF under DMS-0913089.

* Corresponding author.
E-mail addresses: yuqi.wu@colorado.edu (Y. Wu), cai@cs.colorado.edu (X.-C. Cai).

0045-7930/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2010.11.015

The first objective of this paper is to understand the influence of the resistance boundary condition on the accuracy of the numerical simulation. The integral nature of the resistance boundary condition plays an interesting role in the convergence and performance of the Schwarz type preconditioner. In the Newton–Krylov–Schwarz framework, an inexact Newton method with line search is used as a nonlinear solver and within a Newton step, the linear Jacobian system is solved by GMRES preconditioned by a one-level or two-level Schwarz preconditioner. The subdomain partition usually respects the fact that the Jacobian matrix is uniformly sparse and each variable is related through the function only to the neighboring variables, such as in the case of traction boundary condition. However, due to the integral nature of the resistance boundary condition, the Jacobian matrix has a dense block corresponding to all variables on the outlet boundaries. The decomposition of the global domain into subdomains breaks the integral connection between the variables on the outflow boundaries. As far as we know, no one has employed the class of overlapping Schwarz preconditioners to the system with an integral boundary condition. It is worth for us to discuss the impact of the decomposition of the integral condition on the convergence and the scalability of the Schwarz preconditioned Jacobian solver.

We only consider a 2D model in this paper. Although not completely physically realistic for the blood flow simulations, this simplified 2D problem maintains most of the mathematical aspects of the fully coupled fluid–structure interaction problem and also the ability to reproduce the important physical aspects of blood flow in complaint arteries.

The rest of the paper is organized as follows. In Section 2, we describe the formulation of the FSI problem, and also the discretization of the problem, both in space and time. In Section 3, we present the Newton–Krylov–Schwarz method with a two-level Schwarz preconditioner. Finally, in Section 4, we demonstrate the effectiveness of the algorithm by showing some numerical results using different geometries and problem sizes, and report the parallel performance of the algorithm.

2. Mathematical formulation and discretization

With the emphasize in tight and monolithic coupling of the fluid and the artery wall, an additional field is necessary to model the computational mesh in the fluid domain. A popular choice is to model the displacement of fluid mesh by a harmonic extension of the moving fluid–structure boundary. That is, the overall model is described by the coupling of three components, the linear elasticity equation for the wall structure in the reference Lagrangian frame, the incompressible Navier–Stokes equations for the fluid in the ALE framework, and the harmonic equation for the mesh displacement of the fluid domain. See Fig. 1 for a schematic and some notations.

As a whole, the fully coupled FSI problem is given by

$$\rho_s \frac{\partial^2 \mathbf{x}_s}{\partial t^2} - \nabla \cdot \sigma_s - \beta \frac{\partial (\Delta \mathbf{x}_s)}{\partial t} + \gamma \mathbf{x}_s = \mathbf{f}_s \quad \text{in} \quad \Omega_s, \tag{1}$$

$$\mathbf{x}_s = 0 \quad \text{on} \quad \Gamma_s, \tag{2}$$

$$\left.\frac{\partial \mathbf{u}_f}{\partial t}\right|_Y + [(\mathbf{u}_f - \omega_g)\cdot\nabla]\mathbf{u}_f + \frac{1}{\rho_f}\nabla p_f = \nu_f \Delta \mathbf{u}_f + \mathbf{f}_f \quad \text{in} \quad \Omega_f(t), \tag{3}$$

$$\nabla \cdot \mathbf{u}_f = 0 \quad \text{in} \quad \Omega_f(t), \tag{4}$$

$$\mathbf{u}_f = g \quad \text{on} \quad \Gamma_i, \tag{5}$$

$$p_f = RQ = R\int_{\Gamma_o} \mathbf{u}_f \cdot \mathbf{n}\, ds \quad \text{on} \quad \Gamma_o, \tag{6}$$

$$\Delta \mathbf{x}_f = 0 \quad \text{in} \quad \Omega_0, \tag{7}$$

$$\mathbf{x}_f = 0 \quad \text{on} \quad \Gamma_i \cup \Gamma_o. \tag{8}$$

The displacement of the wall $\mathbf{x}_s$ is described by (1) and (2), where $\sigma_s = \lambda_s(\nabla \cdot \mathbf{x}_s)I + \mu_s(\nabla \mathbf{x}_s + \nabla \mathbf{x}_s^T)$ is the Cauchy stress tensor. The Lamé parameters λ_s and μ_s are defined as $\lambda_s = \nu_s E/((1+\nu_s)(1-2\nu_s))$, $\mu_s = E/(2(1+\nu_s))$, where E is the Young's modulus and ν_s the Poisson ratio. ρ_s is the density of the wall structure. β is a visco-elastic damping parameter, and the γ term is used to represent a radially symmetric artery in two dimensions [1].

In order to model the fluid in a moving domain, the fluid mesh displacement $\mathbf{x}_f$ is made to satisfy a harmonic extension of the moving interface as in (7). And a mapping A_t is defined from the reference configuration $\Omega_f(0)$ to the moving domain $\Omega_f(t)$:

$$A_t : \Omega_f(0) \to \Omega_f(t), \quad A_t(\mathbf{Y}) = Y + \mathbf{x}_f(Y), \quad \forall \mathbf{Y} \in \Omega_f(0),$$

where $\mathbf{Y}$ is referred to as the ALE coordinate. The Navier–Stokes equations defined on the moving domain $\Omega_f(t)$ are written in the ALE form (3) and (4), where $\mathbf{u}_f = (u,v)^T$ is the fluid velocity, p_f is the fluid pressure, ν_f is the kinematic viscosity, ρ_f is the fluid density, $\omega_g = \partial A_t/\partial t$ is the velocity of the moving mesh and Y indicates that the time derivative is taken with respect to the ALE coordinates. In (5), a given velocity profile g is prescribed on the inlet boundary Γ_i, while in (6), the resistance boundary condition is prescribed on the outlet boundary Γ_o.

More importantly, three coupling conditions are strongly enforced on the fluid–structure interface Γ_w corresponding to the structure, fluid and moving mesh equations, respectively

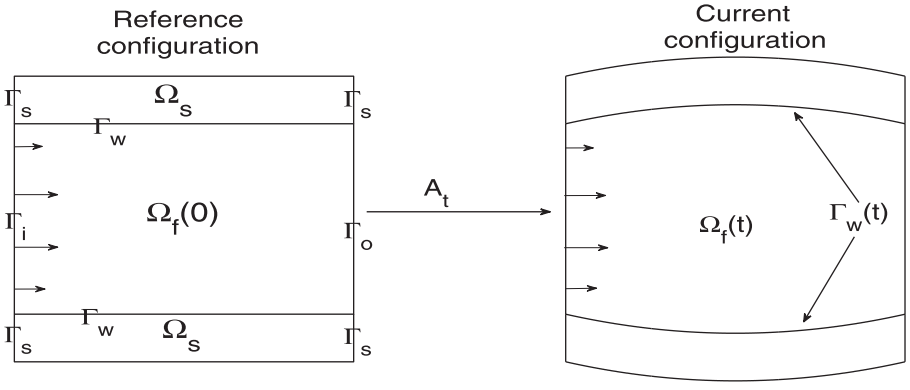

Fig. 1. Ω_s is the structure domain in the Lagrangian reference configuration; $\Omega_f(0)$ is the reference configuration of the fluid domain and $\Omega_f(t)$ represents the moving fluid domain at time t. The inlet and outlet boundaries for the fluid domain Γ_i and Γ_o are fixed and the boundaries Γ_s to the structure domain are also fixed at the inlets and outlets sections, Γ_w represents the fluid–structure interface in the reference configuration.

Please cite this article in press as: Wu Y, Cai X-C. A parallel two-level method for simulating blood flows in branching arteries with the resistive boundary condition. Comput Fluids (2011), doi:10.1016/j.compfluid.2010.11.015

$$\sigma_s \cdot \mathbf{n}_s = -\sigma_f \cdot \mathbf{n}_f, \tag{9}$$

$$\mathbf{u}_f = \frac{\partial \mathbf{x}_s}{\partial t}, \tag{10}$$

$$\mathbf{x}_f = \mathbf{x}_s, \tag{11}$$

where $\mathbf{n}_s$, $\mathbf{n}_f$ are unit normal vectors for the structure and fluid on the fluid–structure interface, σ_s and $\sigma_f = -p_f I + \rho_f \nu_f(\nabla \mathbf{u}_f + \nabla \mathbf{u}_f^T)$ are the Cauchy stress tensors for the structure and fluid, respectively.

By introducing the structure velocity $\dot{\mathbf{x}}_s$ as an additional unknown variable, we can rewrite the structure momentum Eq. (1) to a first order system of equations. We define the variational space of the structure problem as

$$X = \left\{ \mathbf{x}_s \in [H^1(\Omega_s)]^2 : \mathbf{x}_s = 0 \quad \text{on} \quad \Gamma_s \right\}.$$

The weak form of the structure problem is stated as follows: Find $\mathbf{x}_s \in X$ and $\dot{\mathbf{x}}_s \in X$ such that $\forall \phi_s \in X$ and $\forall \varphi_s \in X$,

$$B_s(\{\mathbf{x}_s, \dot{\mathbf{x}}_s\}, \{\phi_s, \varphi_s\}; \sigma_f) = 0,$$

where

$$\begin{aligned} B_s(\{\mathbf{x}_s, \dot{\mathbf{x}}_s\}, \{\phi_s, \varphi_s\}; \sigma_f) = {} & \rho_s \frac{\partial}{\partial t} \int_{\Omega_s} \dot{\mathbf{x}}_s \cdot \phi_s \, d\Omega \\ & + \int_{\Omega_s} \nabla \phi_s : \sigma_s \, d\Omega + \beta \int_{\Omega_s} \nabla \phi_s : \dot{\mathbf{x}}_s \, d\Omega \\ & + \gamma \int_{\Omega_s} \mathbf{x}_s \cdot \phi_s \, d\Omega - \int_{\Gamma_w} \phi_s \cdot (\sigma_f \cdot \mathbf{n}_s) ds \\ & + \int_{\Omega_s} \left(\frac{\partial \mathbf{x}_s}{\partial t} - \dot{\mathbf{x}}_s \right) \cdot \varphi_s \, d\Omega - \int_{\Omega_s} \mathbf{f}_s \cdot \phi_s \, d\Omega. \end{aligned}$$

The structure problem B_s depends on $\mathbf{u}_f$ and p_f by the coupling condition (9). Since this coupling condition is defined on the fluid–structure interface, the structure problem also depends on the fluid mesh displacement $\mathbf{x}_f$ implicitly.

The variational spaces of the fluid subproblem are time-dependent, and the solution of the structure subproblem provides an essential boundary condition for the fluid subproblem by (10). We define the trial and weighting function spaces as:

$$V = \left\{ \mathbf{u}_f \in [H^1(\Omega_f(t))]^2 : \mathbf{u}_f = g \quad \text{on} \quad \Gamma_i, \mathbf{u}_f = \frac{\partial \mathbf{x}_s}{\partial t} \quad \text{on} \quad \Gamma_w \right\},$$

$$V_0 = \left\{ \mathbf{u}_f \in [H^1(\Omega_f(t))]^2 : \mathbf{u}_f = 0 \quad \text{on} \quad \Gamma_i \cup \Gamma_w \right\},$$

$$P = L^2(\Omega_f(t)).$$

The weak form of the fluid problem reads: Find $\mathbf{u}_f \in V$ and $p_f \in P$ such that $\forall \phi_f \in V_0$ and $\forall \psi_f \in P$,

$$B_f(\{\mathbf{u}_f, p_f\}, \{\phi_f, \psi_f\}; \mathbf{x}_f) = 0,$$

where

$$\begin{aligned} B_f(\{\mathbf{u}_f, p_f\}, \{\phi_f, \psi_f\}; \mathbf{x}_f) = {} & \int_{\Omega_f(t)} \left. \frac{\partial \mathbf{u}_f}{\partial t} \right|_Y \cdot \phi_f \, d\Omega - \int_{\Omega_f(t)} \mathbf{f}_f \cdot \phi_f \, d\Omega \\ & + \int_{\Omega_f(t)} [(\mathbf{u}_f - \omega_g) \cdot \nabla] \mathbf{u}_f \cdot \phi_f \, d\Omega \\ & + \nu_f \int_{\Omega_f(t)} \nabla \mathbf{u}_f : \nabla \phi_f \, d\Omega \\ & - \int_{\Omega_f(t)} p_f (\nabla \cdot \phi_f) d\Omega + \int_{\Omega_f(t)} (\nabla \cdot \mathbf{u}_f) \psi_f \, d\Omega \\ & - \int_{\Gamma_o} \mathbf{t}_f \cdot \phi_f \, ds. \end{aligned} \tag{12}$$

The last term in (12), which represents the contribution of the resistance boundary condition on the outlet Γ_o, can be rewritten as

$$\int_{\Gamma_o} \mathbf{t}_f \cdot \phi_f \, ds = -\int_{\Gamma_o} \left(R \int_{\Gamma_o} \mathbf{u}_f \cdot \mathbf{n} ds \right) \phi_f \cdot \mathbf{n} ds + \nu_f \int_{\Gamma_o} \phi_f \cdot (\nabla \mathbf{u}_f) \cdot \mathbf{n} ds$$

thanks to the relation (6), where the pseudo-stress tensor $\mathbf{t}_f = -p\mathbf{n} + \nu_f \nabla \mathbf{u}_f \cdot \mathbf{n}$. If the zero-traction boundary condition is considered, the pseudo-stress tensor $\mathbf{t}_f = 0$ on Γ_o, and the last term in (12) can be omitted. The fluid problem couples the fluid mesh displacement $\mathbf{x}_f$ through the term ω_g and the integration domain, and connects to the structure velocity $\dot{\mathbf{x}}_s$ by the coupling condition (10).

The weak form of the mesh movement problem reads: Find $\mathbf{x}_f \in Z$ such that $\forall \xi \in Z_0$

$$B_m(\mathbf{x}_f, \xi) = 0,$$

where

$$B_m(\mathbf{x}_f, \xi) = \int_{\Omega_0} \nabla \xi : \nabla \mathbf{x}_f \, d\Omega.$$

And the variational spaces are defined as

$$Z_0 = \left\{ \mathbf{x}_f \in [H^1(\Omega_0)]^2 : \mathbf{x}_f = 0 \quad \text{on} \quad \Gamma_i \cup \Gamma_o \cup \Gamma_w \right\},$$

$$Z = \left\{ \mathbf{x}_f \in [H^1(\Omega_0)]^2 : \mathbf{x}_f = \mathbf{x}_s \quad \text{on} \quad \Gamma_w, \mathbf{x}_f = 0 \quad \text{on} \quad \Gamma_i \cup \Gamma_o \right\}.$$

We use unstructured LBB-stable mixed Q2–Q1 elements for the fluid and Q2 elements for the structure. Denote the finite element subspaces X_h, V_h, $V_{h,0}$, P_h, Z_h, $Z_{h,0}$ as the counterparts of their infinite dimensional subspaces. We form the finite dimensional fully coupled FSI problem as: Find $x_s \in X_h$, $\dot{x}_s \in X_h$, $u_f \in V_h$, $p_f \in P_h$ and $x_f \in Z_h$ such that $\forall \phi_s \in X_h$, $\forall \varphi_s \in X_h$, $\forall \phi_f \in V_{h,0}$, $\forall \psi_f \in P_h$, and $\forall \xi \in Z_{h,0}$,

$$B_s(\{x_s, \dot{x}_s\}, \{\phi_s, \varphi_s\}; \sigma_f) + B_f(\{u_f, p_f\}, \{\phi_f, \psi_f\}; x_f) + B_m(x_f, \xi) = 0. \tag{13}$$

The fully coupled system (13) is further discretized in time with a second-order trapezoidal rule. Since the temporal discretization scheme is fully implicit, at each time step, we obtain the solution x^n at the n_{th} time step from the previous time step by solving a sparse, nonlinear algebraic system

$$\mathscr{F}_n(x^n) = 0,$$

where the vector x^n corresponds to the nodal values of the fluid velocity $\mathbf{u}_f$, the fluid pressure p_f, the fluid mesh displacement $\mathbf{x}_f$, the structure displacement $\mathbf{x}_s$ and the structure velocity $\dot{\mathbf{x}}_s$ at the n_{th} time step. For simplicity, we ignore n for the rest of the paper. Equivalently, at each time step, we solve the nonlinear system

$$\mathscr{F}(x) = 0, \tag{14}$$

to obtain the solution x.

3. Newton–Krylov–Schwarz methods with a two-level preconditioner

The fully discretized system (14) is nonlinear. The nonlinearity comes through the convection term of the Navier–Stokes equations and the displacement of the fluid domain. In the Newton–Krylov–Schwarz approach, the nonlinear system is solved via the inexact Newton method [11,12]. At each Newton step the new solution $x^{(k+1)}$ is updated by the current solution $x^{(k)}$ and the Newton correction $s^{(k)}$,

$$x^{(k+1)} = x^{(k)} + \theta^{(k)} s^{(k)},$$

where the step length parameter $\theta^{(k)}$ is determined by a cubic line search technique [10], and $x^{(0)}$ is the solution from the previous time step. The Newton correction $s^{(k)}$ is approximated by solving a preconditioned Jacobian system with a Krylov subspace method, GMRES [21].

Please cite this article in press as: Wu Y, Cai X-C. A parallel two-level method for simulating blood flows in branching arteries with the resistive boundary condition. Comput Fluids (2011), doi:10.1016/j.compfluid.2010.11.015

$$J_k M_k^{-1} M_k s^{(k)} = -\mathscr{F}(x^{(k)}), \tag{15}$$

where J_k is the Jacobian matrix evaluated at $x^{(k)}$. The accuracy of approximation is controlled by the forcing term η_k to ensure the residual of (15) is decreased by some desired tolerance

$$\left\| \mathscr{F}(x^{(k)}) + J_k M_k^{-1} M_k s^{(k)} \right\| \leqslant \eta_k \left\| \mathscr{F}(x^{(k)}) \right\|.$$

Since the parallel efficiency and the parallel scalability are mostly determined by the preconditioning of the Jacobian system (15), we next describe the construction of one-level and two-level Schwarz preconditioners in detail.

To define the additive Schwarz method, we first partition the finite element mesh $\mathscr{T}^h = \Omega_f^h \cup \Omega_s^h$ into non-overlapping subdomains $\Omega_\ell^h, \ell = 1, \ldots, N$, without using the information of the location of the interface. Each one is distributed to a processor of the parallel machine. To insure that this partition with good load balancing, the package ParMETIS [15] is employed to decompose the mesh into parts with fairly equal number of unknowns. Then, the subdomains Ω_ℓ^h are extended to overlapping domains $\Omega_\ell^{h,\delta}$, where δ represents the size of overlap in terms of layers of elements. The so-called one-level restricted additive Schwarz (RAS) preconditioner [7] is defined by

$$M_{one}^{-1} = \sum_{\ell=1}^{N} (R_\ell^0)^T J_\ell^{-1} R_\ell, \tag{16}$$

where R_ℓ^0 and R_ℓ are restrictions to the degrees of freedom in the non-overlapping subdomain Ω_ℓ^h and overlapping subdomain $\Omega_\ell^{h,\delta}$, respectively. The subdomain operator J_ℓ is a restriction of the Jacobian matrix, defined by $J_\ell = R_\ell J R_\ell^T$. The local problems are solved by the LU factorization.

The one-level method is most effective when the number of processors is small. It is necessary to include a global coarse mesh to provide global information exchange between the subdomains in order to obtain fast convergence when the number of processors is large.

To formally define the hybrid two-level Schwarz preconditioner, we first specify the choice of the coarse mesh and the partition based on that. In our setting, the fine mesh does not have to be the refinement of the coarse mesh. The information exchange between the coarse mesh and the fine mesh is through the coarse-to-fine interpolation matrix I_H^h. The coarse mesh inherits the partition of the fine mesh, see Fig. 2 for an example. The construction of the interpolation matrix I_H^h is based on the finite element interpolation. For each degree of freedom v_i on the fine mesh, it is extrapolated from the coarse mesh by

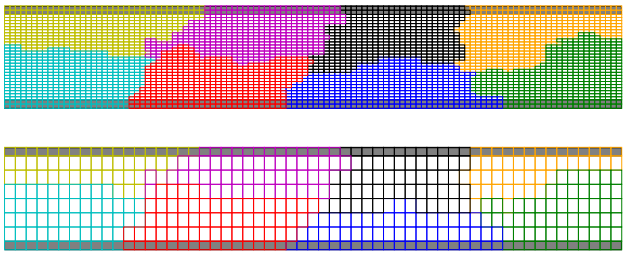

Fig. 2. Example partitions of the mesh of a straight tube case by using ParMETIS. The subdomains are marked with different colors and the structure elements are shaded. The top figure shows the partition of the fine mesh, and the bottom represents a corresponding partition of the coarse mesh. Both of the partitions are based on the elements, and the structure elements and fluid elements can be grouped into the same subdomain. The elements of a coarse subdomain are assigned to the corresponding processor that contains the fine mesh elements from the same subdomain.

$$v_i \approx (I_H^h v^H)_i = \sum_j v_j^H \phi_j^H(x_i),$$

where x_i is the coordinates associated with the degree of freedom v_i on the fine mesh, v^H is a function on the coarse mesh, and ϕ_j^H is the finite element basis function of the jth degree of freedom on the coarse mesh. In our simulation, the fluid mesh deforms through the fluid–structure interaction, the interpolation matrix should be updated at each time step. Since the calculation of the basis function is costly, for simplicity, we construct the interpolation matrix only based on the reference configuration.

We combine the coarse level preconditioner M_H^{-1} and the fine level preconditioner M_h^{-1} multiplicatively. In other words, the application of the two-level hybrid Schwarz preconditioner $y = M_{two}^{-1} x$ is obtained in the following two steps

$$y = I_H^h M_H^{-1} (I_H^h)^T x, \tag{17}$$

$$M_{two}^{-1} x = y + M_h^{-1}(x - J_h y), \tag{18}$$

where J_h is the Jacobian matrix obtained on the fine mesh. We choose the fine level preconditioner M_h^{-1} as the one-level restricted additive Schwarz preconditioner given in (16). The coarse level preconditioning by M_H^{-1} requires solving a linear system on the coarse mesh, which is often computationally costly if the linear system is solved directly. Therefore, we solve the coarse problem iteratively by using GMRES with a one-level restricted additive Schwarz preconditioner. Since the coarse solution is approximated by an iterative method, the two-level preconditioner changes in each linear iteration. In this case, the flexible GMRES (fGMRES) is more appropriate than the regular GMRES [22].

To complete this section, we make a remark on the impact of the decomposition into subdomains on the resistance boundary condition. The only criterion we use when partition the domain into subdomains is to minimize the edge cuts; i.e., the outlet boundary would be cut into different subdomains, see Fig. 2. In such a situation, the global connection between the variables on the outlet boundaries, which is established by the resistance boundary condition (6), is destroyed. In the next section, we will discuss the convergence and parallel performance of the Schwarz type preconditioner with the resistive boundary condition.

4. Numerical results

In this section, we report some numerical results of the proposed FSI solver by simulating the blood flows in compliant arteries. At first, we validate our model by studying the impact of different outflow boundary conditions on the accuracy of the hemodynamic prediction. We then investigate the numerical behavior and parallel performance of the two-level hybrid Schwarz preconditioner with the resistive boundary condition. The software is developed using the Portable Extensible Toolkit for Scientific computation (PETSc) library [2]. Mesh generations are carried out by Cubit [19]. The tests reported in this section are performed on an IBM BlueGene/L with 4096 compute nodes.

4.1. Impact of different outflow boundary conditions

We first study the application of our method to a two-dimensional model of the artery with a simple geometry. The model consists of a straight tube of 20 cm in length, 2 cm in diameter and the artery wall is 0.2 cm in thickness. For the inlet, we prescribe a pulsatile periodic flow wave,[1] with a period T of 0.6 s, mapped to a uniform velocity profile, see Fig. 3. For the outlet, both the

[1] The data is provided by Z. Su, K. Hunter, and R. Shandas of University of Colorado, School of Medicine.

Please cite this article in press as: Wu Y, Cai X-C. A parallel two-level method for simulating blood flows in branching arteries with the resistive boundary condition. Comput Fluids (2011), doi:10.1016/j.compfluid.2010.11.015

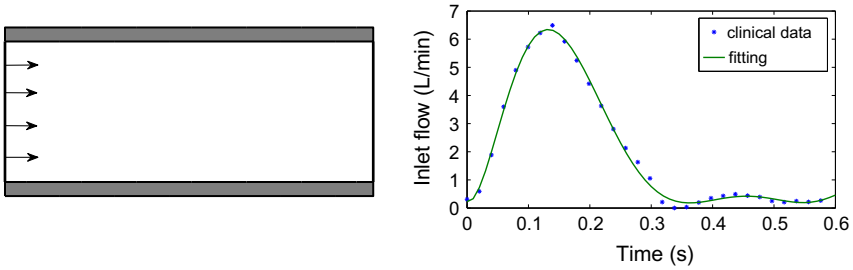

Fig. 3. The setup of the straight tube problem and the inlet flow rate from clinical data with a polynomial fitting.

zero-traction and resistance boundary conditions are considered. In the case of the resistance boundary condition, the resistance $R = 599.85$ dyn·s/cm^5 is prescribed on the outlet. The elastic artery wall is characterized with density of 1.2 g/cm^3, Young's modulus of 6.0×10^6 g/(cm s^2) and Poisson ratio of 0.48. The blood is modeled as a Newtonian flow, with a density of 1.0 g/cm^3 and kinematic viscosity of 0.035 cm^2/s. The Reynolds number is approximately 300 based on the mean inlet velocity and the radius of the artery. It is important that the simulation starts from a equilibrium state [13]. Therefore, we obtain the initial condition as follows:

1. The initial velocity of the wall $\dot{x}_{s,0}$ is set to zero.
2. A steady FSI problem is solved, using the same inlet and outlet boundary conditions as in the time-dependent problem. The velocity $u_{f,0}$, the pressure $p_{f,0}$, the fluid mesh displacement $x_{f,0}$, the structure displacement $x_{s,0}$ are then used as the initial conditions for the unsteady problem.

A mesh with 58,369 elements and 1.01 million degrees of freedom is utilized and the solutions are obtained using a time step of 1 ms for a total of 3 cardiac cycle. The simulation proceeds to the next time step when the residual of the nonlinear system is less than 10^{-6}. In Fig. 4, we compare the results obtained with two types of outflow boundary conditions, the resistance and the zero-traction boundary conditions. For the resistance boundary condition, the wall deformation alters the flow distributions at the inlet and outlet sections over one cardiac cycle. Because the compliant vessels store flow during the systole phase and then release flow during the diastole phase, the outlet flow rate is smaller compared to the inlet flow rate at the peak systole phase, while the outlet flow rate is larger at the diastole. It is clear that the wall displacement plot follows the shape of the wall pressure plot, which shows that the artery walls move in response to the pulse of the pressure. For the zero-traction boundary condition, both plots are dramatically different from the resistance case. The outlet flow shows unrealistic peak and negative amplitude, and the wall pressure and displacement results in unrealistic amplitude and oscillatory pattern.

To further illustrate the influence of the boundary conditions, we carry out numerical simulations for a model artery bifurcating to two branches. As shown in Fig. 5, one of the branches stenoses with a 75% area reduction. We use the same inlet boundary condition, and the same material properties for the blood and vessel wall as in the previous test. The finite element mesh consists of 65,920 elements and 1.15 million of unknowns. The simulations are run for 3 cardiac cycles with a time step size of 1 ms. Fig. 6 depicts the flow waves as well as the mean flow at outlets during one cardiac cycle for the two different boundary conditions. With the resistance boundary condition, the mean flow split between the two branches nearly half by half, a bit more flow goes to the

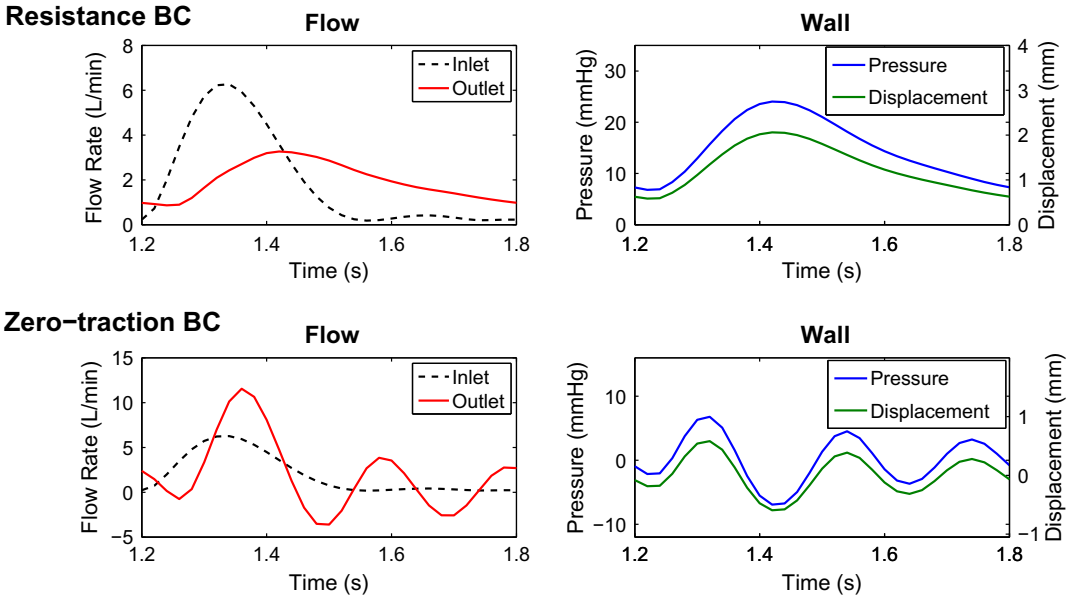

Fig. 4. Flow waves at inlet and outlet, wall pressure and displacement over one cardiac cycle, obtained using resistance and zero-traction outlet boundary conditions. Top left figure represents the inlet and outlet flow rate, and top right figure shows the wall displacement and pressure by using the resistance boundary condition, while the bottom figures represent the results by using the zero-traction outflow boundary condition.

Please cite this article in press as: Wu Y, Cai X-C. A parallel two-level method for simulating blood flows in branching arteries with the resistive boundary condition. Comput Fluids (2011), doi:10.1016/j.compfluid.2010.11.015

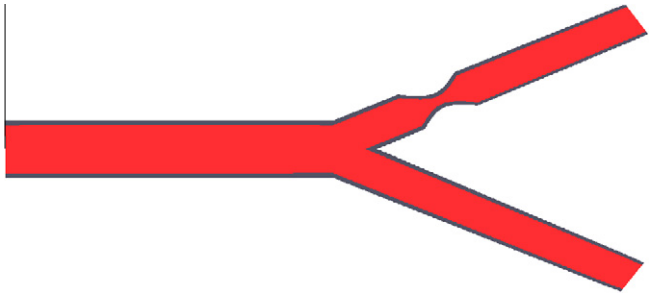

Fig. 5. Geometric details of the bifurcating artery, a 75% area reduction stenosis is shown on one of the branches.

normal artery branch than the stenosed one during the systole phase. For the zero-traction boundary condition, the normal artery branch receives more flow than the stenosed branch throughout the cardiac cycle and unrealistic negative flow is observed at the outlet of the stenosed branch. As a result, the mean flow split in a unrealistic pattern, nearly 90% of the flow goes to the normal branch. In fact, it has been verified in clinical experiments that the flow distributes almost evenly (50–50%) between the normal branch and the stenosed branch, provided that the stenosis is less than 85% area reduction [25]. In this sense, the simulation obtained with the resistance boundary condition shows a better consistency with the clinical results.

Fig. 7 provides an illustration of the differences in the fluid velocity and pressure fields between the resistance and zero-traction outflow boundary conditions at peak systole. Note that the differences are not small.

Fig. 8 shows the computed wall shear stress and wall displacement at different locations along the artery wall during one cardiac cycle by using the resistance outflow boundary condition. We place the first measurement location P0 at the middle point of the upstream artery wall. On the wall of the stenosed branch, P1 and P5 are placed a distance of 1.0 cm from the throat of the stenosis. P3 is located at the throat of the stenosis, and P7 is placed near the outlet boundary. On the wall of the normal branch, P2, P4, P6, P8 represent the corresponding locations to those specified locations on the wall of the stenosed branch. On the wall of the stenosed branch, the highest wall shear stress occurs at the throat of stenosis (P3). The wall shear stress at P3 increases rapidly in the systole phase, and decreases in the diastole phase. At location P1, the wall shear stress shows a little fluctuation during the cardiac cycle. It increases in the systole phase, and returns to a constant in the diastole phase. At location P5, the wall shear stress shows a more oscillatory pattern. It increases in the systole phase, and oscillates in the early diastole phase. Note that, in the stenosis section, the fluid can accelerate to a high speed and produce a recirculation zone in the downstream of the stenosis. At location P7, which is away from the recirculation zone and closer to the outflow boundary, the wall shear stress pattern is closer to the one at location P1. On the wall of the normal branch, at the locations P2, P4, P6, the wall shear stress are almost the same at any time during the cardiac cycle, increasing in the systole phase and decreasing in the diastole phase. At location P8, the wall shear stress behaves a little differently. The peak wall shear stress is smaller and decreases slower in the diastole phase. As for the wall displacement, on the stenosed branch, on average, the wall displacement is smaller at the stenosis throat (P3), and is larger at the location P7, which is located near the outflow boundary. On average, the wall displacement at P8 is larger than the wall displacement at other locations on the normal branch. The wall displacement at P2, P4, and P6 share a similar pattern and magnitude.

4.2. Performance and parallel scalability

In this subsection, we discuss the performance and parallel scalability of the two-level solver to the fully coupled FSI problem using the resistance boundary condition. For all the numerical tests in this subsection, unless otherwise specified, we use the same geometries, material properties and boundary conditions as

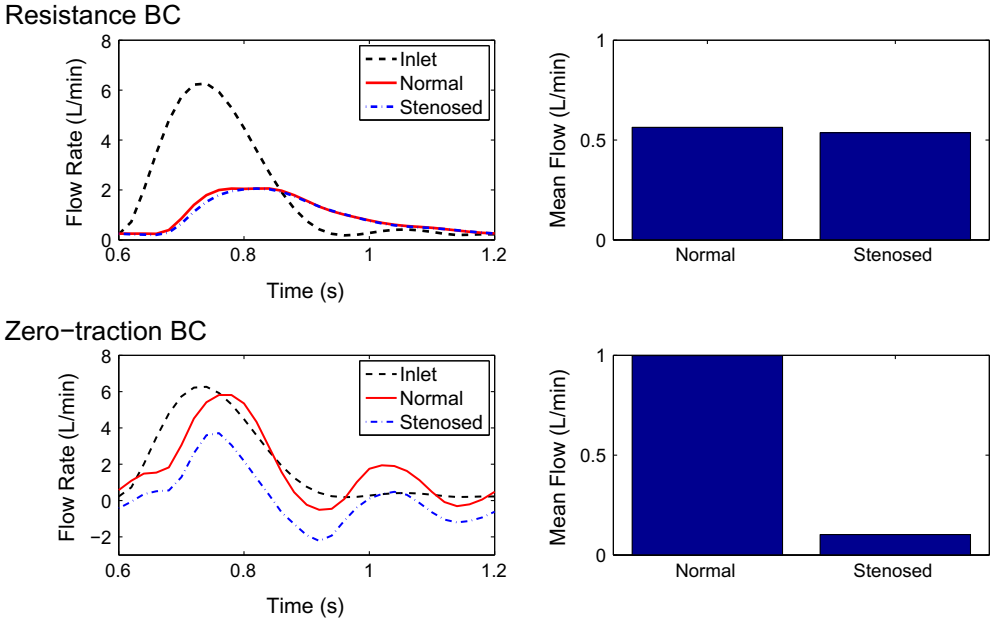

Fig. 6. Outlet flow waves and flow distribution between the normal artery branch and the stenosed artery branch with a 75% area reduction stenosis during one cardiac cycle, by using the resistance and zero-traction outflow boundary condition. Top left plot shows the flow rate at the inlet and outlet boundary during one cardiac cycle by using the resistance boundary condition, and the top right plot gives the results of the flow distribution between the normal artery branch and stenosed branch by using the resistance boundary condition. Corresponding results by using the zero-traction boundary conditions are shown in the bottom plots.

Please cite this article in press as: Wu Y, Cai X-C. A parallel two-level method for simulating blood flows in branching arteries with the resistive boundary condition. Comput Fluids (2011), doi:10.1016/j.compfluid.2010.11.015

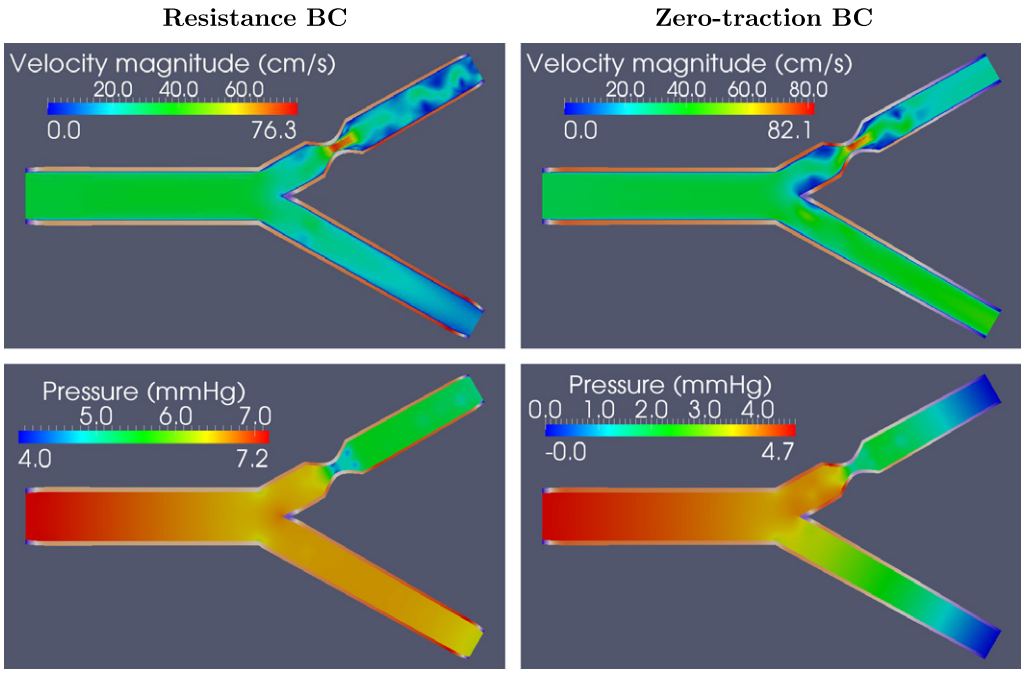

Fig. 7. Fluid velocity magnitude and pressure at peak systole for resistance and zero-traction outflow boundary conditions. The artery walls are colored by the magnitude of the structural displacement. For the resistance boundary condition, fluid velocity magnitude and pressure are shown on the left. Corresponding figures using the zero-traction boundary are shown on the right.

described before. The stopping criterion for the Newton iteration is when the norm of the residual of the nonlinear system is less than 10^{-6}. The accuracy of the preconditioned Jacobian system is governed by the relative tolerance of 10^{-4}. The coarse solver is considered to have converged if the relative residual is less than 10^{-3}. The time step size is fixed as $\Delta t = 1$ ms, and the simulation is stopped after 10 time steps. We then report the results, such as the average compute time, the average number of Newton iterations per time step, and the average fGMRES iterations per Newton step.

4.2.1. Straight tube case

We consider the benchmark straight tube problem as described before. We first demonstrate the performance of the hybrid two-level preconditioner with the resistance boundary condition, by comparing to the results obtained with the zero-traction boundary condition. In [4], the performance of the preconditioner with the zero-traction boundary condition was studied. As observed in Table 1, the two-level hybrid preconditioner handles the resistance boundary condition well. In terms of the number of iterations and the average compute time, the preconditioner shows good performance with both boundary conditions. As we double the number of processors, the number of linear iterations keeps close to a constant while the compute time is cut by half in both cases. Although, in the case of the resistance boundary condition, the subdomain partition breaks the integral connection between the variables on the outlet boundary, the two-level hybrid preconditioner still proves to be an effective choice.

For the one-level additive Schwarz preconditioner, the preconditioned system becomes more ill-conditioned as the number of subdomains increases. In Table 2, we show the results obtained with the two-level preconditioner as we increase the number of subdomains. Comparing to the results of the one-level preconditioner, the performance of the two-level preconditioner is much better. As we increase the number of subdomains to 1024, the average number of linear iterations per Newton step stays close to a constant. Similar results are observed in terms of the average compute time.

In order to understand the parallel scalability of the two-level preconditioner, we show the parallel speedup over the number of processors for the two different meshes. As shown in Fig. 9, for both problems, the two-level preconditioner shows nearly ideal speedup with up to 1024 processors. We are also interested in the weak scalability of the algorithms. Ideally, as we increase the number of processors and the problem size at the same rate, the number of linear iterations and the compute time should not change much. In this sense, the two-level preconditioner shows much better weak scalability than the one-level preconditioner, especially when the number of processors is large, as shown in Fig. 10.

There are several important factors that impact the performance of the two-level preconditioner, such as the interpolation operator, the tolerance of the coarse solve. The motivations for two-level preconditioner is to facilitate the exchange of information between subdomains by adding a coarse mesh, thereby improve the performance of the preconditioning. In practice, a relatively fine coarse mesh would help reduce the total number of iterations, but the overall compute time may increase. The best choice of the coarse mesh needs to balance the time of the coarse solve and the preconditioning effects. In Table 3, we show some results obtained with different coarse mesh sizes.

4.2.2. Branching artery case

In this subsection, we consider the a branching artery model problem. One of the branches of the artery stenoses with a 75% area reduction. In this geometry, the computations is more challenging because relatively high Reynolds numbers occur at and near the stenosis. In Table 4, we compare the two-level preconditioner with the one-level preconditioner. Again, the two-level preconditioner is more effective than the one-level preconditioner in terms of the average number of linear iterations and the average compute time.

As for the parallel scalability, we show the speedup of the two-level preconditioner in Fig. 11. The two-level preconditioner shows nearly ideal speedup with up to 1024 processors. The effects of coarse mesh size are shown in Table 5. Comparing to the one-level

Please cite this article in press as: Wu Y, Cai X-C. A parallel two-level method for simulating blood flows in branching arteries with the resistive boundary condition. Comput Fluids (2011), doi:10.1016/j.compfluid.2010.11.015

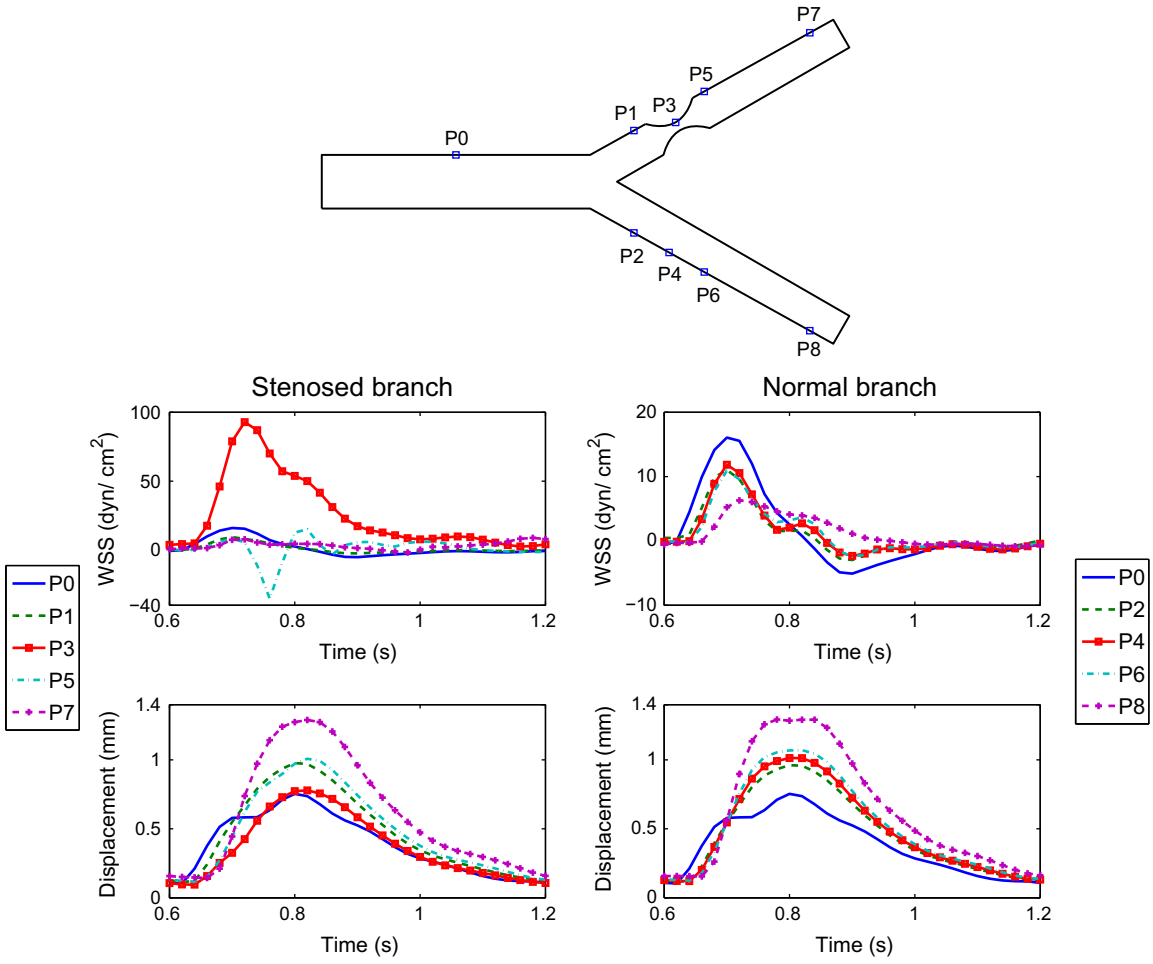

Fig. 8. Wall shear stress (WSS) and wall displacement at different locations of the artery wall during one cardiac cycle, by using the resistance outflow boundary condition. Top plot shows the locations of measurement on the artery wall. Middle left plot shows the WSS at the locations (P1, P3, P5, P7) of the stenosed branch during one cardiac cycle, and middle right plot shows the WSS at the corresponding locations (P2, P4, P6, P8) of the normal branch during one cardiac cycle. Since we use different scale for the WSS on the two different branches, we include the WSS at the upstream location P0 (the solid line) in both plots for reference. The bottom two plots show the wall displacement for those locations on the stenosed branch and the normal branch, respectively, from left to right.

Table 1
Comparison of the performance of the two-level hybrid preconditioner with the zero-traction and resistance outflow boundary conditions. The tests are carried on a straight tube problem, associated with the fine mesh of 2.01×10^6 degrees of freedom and the coarse mesh of 1.30×10^5 degrees of freedom. "*np*" denotes the number of processors. "Newton" denotes the average Newton iterations per time step. "fGMRES" denotes the average fGMRES iterations per Newton step. "Time" refers the average compute time, in seconds, per time step.

np	Zero-traction			Resistance		
	Newton	fGMRES	Time (s)	Newton	fGMRES	Time (s)
64	2.0	36.96	210.27	2.0	38.18	214.31
128	2.0	39.59	83.17	2.0	40.46	84.43
256	2.0	46.96	41.32	2.0	48.59	42.76
512	2.0	52.68	22.53	2.0	54.09	23.33

Table 2
Performance of the two different preconditioners with respect to the increasing number of subdomains for the straight tube test problem. The tests are carried on a mesh with 2.01×10^6 unknowns, using the resistance outflow boundary condition. "Newton" denotes the average Newton iterations per time step. "fGMRES" denotes the average fGMRES iterations per Newton step. "Time" denotes the average compute time, in seconds, per time step.

np	One-level			Two-level		
	Newton	fGMRES	Time (s)	Newton	fGMRES	Time (s)
64	2.0	103.91	236.47	2.0	38.18	214.31
128	2.0	124.09	107.29	2.0	40.46	84.43
256	2.0	218.06	67.07	2.0	48.59	42.76
512	2.0	281.55	44.03	2.0	54.09	23.33
1024	2.0	450.68	36.17	2.0	75.32	17.34

preconditioner, the two-level preconditioner greatly reduces the average number of linear iterations as well as the average compute time.

In the one-level preconditioner, the overlap parameter δ is often used to control the amount of information exchange between the subdomains. Larger overlaps allow more information to exchange, but require additional communication time. Therefore the preconditioner improves the condition number of the linear system but spends more time in communication. For the two-level preconditioner, the exchange of information happens through the interpolation from the coarse mesh. In other words, the two-level approach is less sensitive to the overlap parameter. In practice, the one-level preconditioner fails to converge for zero overlap, while the two-level preconditioner performs well. Results with respect to the overlaps are shown in Table 6.

4.3. Robustness to parameters

In the FSI simulation, the elastic artery wall is characterized by two important parameters, the Young's modulus and the Poisson

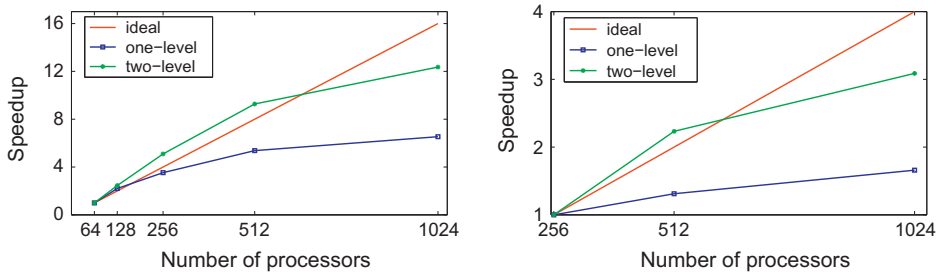

Fig. 9. Speedup versus number of processors for straight tube problem with the resistance outflow boundary condition. Results for the problem with 2.01 millions of unknowns are on the left, and results for the problem with 3.99 millions of unknowns show on the right.

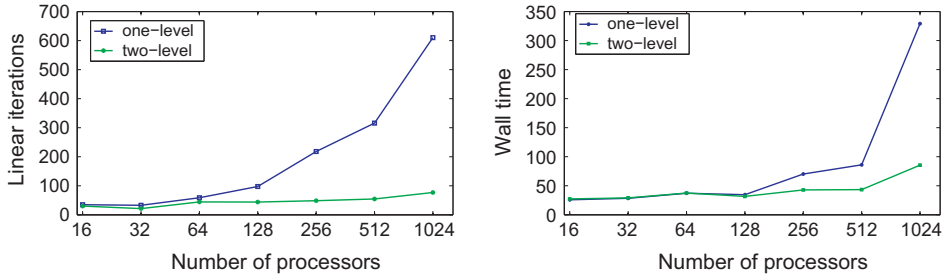

Fig. 10. Weak scaling of the algorithm for the straight tube problem using the resistance boundary condition. On the left, the vertical axis shows the average number of linear iterations per Newton step. On the right, the vertical axis shows the average compute time in seconds per time step. The number of unknowns increases with the number of processors: 2.01×10^6 for 256 processors, 7.99×10^6 for 1024 processors.

Table 3
The effect of various choices of the preconditioners for the straight tube problem with different problem sizes and number of processors. The heading "coarse size" represents the number of unknowns on the coarse mesh as a fraction of the number of unknowns on the fine mesh and "coarse time" is the time spent on the coarse solve as a fraction of total compute time.

Unknowns	*np*	Levels	Coarse size	Newton	fGMRES	Time (s)	Coarse time
2.01×10^6	256	One	0.0	2.0	218.05	70.14	0.0
2.01×10^6	256	Two	0.02	2.0	86.23	42.08	0.18
2.01×10^6	256	Two	0.03	2.0	73.46	43.03	0.23
2.01×10^6	256	Two	0.06	2.0	48.59	42.76	0.28
2.01×10^6	512	One	0.0	2.0	281.55	44.03	0.0
2.01×10^6	512	Two	0.02	2.0	94.68	23.11	0.33
2.01×10^6	512	Two	0.03	2.0	82.55	23.91	0.38
2.01×10^6	512	Two	0.06	2.0	54.09	23.33	0.42
3.99×10^6	512	One	0.0	2.0	315.91	86.01	0.0
3.99×10^6	512	Two	0.01	2.0	109.82	46.88	0.19
3.99×10^6	512	Two	0.03	2.0	55.32	43.16	0.26
3.99×10^6	512	Two	0.06	2.0	49.05	44.95	0.30
3.99×10^6	512	Two	0.13	2.0	31.86	61.98	0.54
7.99×10^6	1024	One	0.0	2.0	610.22	329.20	0.0
7.99×10^6	1024	Two	0.01	2.0	189.64	76.73	0.29
7.99×10^6	1024	Two	0.03	2.0	121.36	81.63	0.44
7.99×10^6	1024	Two	0.06	2.0	76.86	85.43	0.52

Table 4
Performance of two different precondtioners with increasing number of subdomains for the branching artery problem. The tests are carried on a mesh with 2.00×10^6 unknowns, using the resistance outflow boundary condition.

np	One-level			Two-level		
	Newton	fGMRES	Time (s)	Newton	fGMRES	Time (s)
128	2.0	124.32	158.40	2.0	56.14	118.59
256	2.0	177.55	69.80	2.0	51.23	45.29
512	2.0	403.14	56.81	2.0	86.73	27.68
1024	2.0	556.71	43.45	2.0	104.86	17.46

ratio, which characterize the stiffness and the incompressibility of the elastic wall, respectively. The stiffer and the larger Poisson ratio the wall are, the more difficulty it is to solve the problem numerically. In Table 7, the two-level preconditioner shows robust convergence with respect to the Young's modulus. In Table 8, the two-level preconditioner also shows robust performance with respect to the Poisson ratio. Increasing the Poisson ratio has only moderate effect to the solver. Other important parameters in the simulations are the fluid density and the wall structure density. It has been reported in some references that the convergence becomes more difficult to achieve if the density of the fluid and the wall structure are close to each other [9], or if the fluid is much more heavier than the wall structure [17]. In Table 9, the two-level solver performs fairly well for a wide range of fluid density and wall structure density. For the resistance boundary condition, the value of the resistance is one of the important parameter in the simulation. In Table 10, the two-level solver shows robust performance with respect to the value of resistance.

Please cite this article in press as: Wu Y, Cai X-C. A parallel two-level method for simulating blood flows in branching arteries with the resistive boundary condition. Comput Fluids (2011), doi:10.1016/j.compfluid.2010.11.015

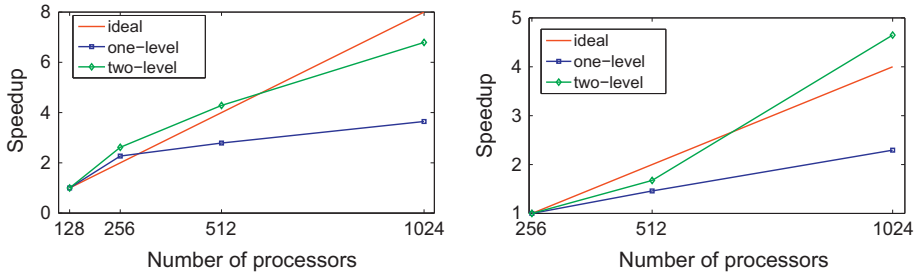

Fig. 11. Speedup versus number of processors for the branching artery model with the resistance outflow boundary condition. Results for the problem with 2.00 millions of unknowns is on the left, and results for the problem with 3.88 millions of unknowns show on the right.

Table 5
The effect of various choices of the size of the coarse mesh for the branching artery problem with different problem sizes and number of processors. "coarse size" represents the number of unknowns on the coarse mesh as a fraction of the number on the fine mesh and "coarse time" is the time spent on the coarse solve as a fraction of total compute time.

Unknowns	*np*	Levels	Coarse size	Newton	fGMRES	Time (s)	Coarse time
9.82×10^5	128	One	0.0	2.0	83.77	40.77	0.0
9.82×10^5	128	Two	0.02	2.0	69.86	36.41	0.15
9.82×10^5	128	Two	0.04	2.0	62.82	36.86	0.27
9.82×10^5	128	Two	0.07	2.0	49.23	37.40	0.21
2.00×10^6	256	One	0.0	2.0	177.55	69.80	0.0
2.00×10^6	256	Two	0.02	2.0	87.14	44.43	0.17
2.00×10^6	256	Two	0.03	2.0	76.82	46.77	0.23
2.00×10^6	256	Two	0.07	2.0	51.23	45.29	0.26
3.88×10^6	512	One	0.0	2.0	219.50	107.15	0.0
3.88×10^6	512	Two	0.01	2.0	124.77	77.70	0.14
3.88×10^6	512	Two	0.03	2.0	164.18	80.94	0.19
3.88×10^6	512	Two	0.06	2.0	95.23	70.67	0.28
3.88×10^6	1024	One	0.0	2.0	381.59	68.18	0.0
3.88×10^6	1024	Two	0.01	2.0	166.41	33.39	0.38
3.88×10^6	1024	Two	0.03	2.0	105.64	27.82	0.36

Table 6
The effect of the overlap parameter for one-level and two-level preconditioners for the branching case with the resistance outflow boundary condition.

Unknowns	*np*	Levels	Overlap	Newton	fGMRES	Time (s)
9.82×10^5	128	One	1	2.0	132.05	43.28
9.82×10^5	128	One	2	2.0	83.77	40.77
9.82×10^5	128	One	3	2.0	60.91	43.42
9.82×10^5	128	Two	0	2.0	69.86	36.41
9.82×10^5	128	Two	1	2.0	65.05	42.12
9.82×10^5	128	Two	2	2.0	49.46	41.50
2.00×10^6	256	One	1	2.0	471.09	78.16
2.00×10^6	256	One	2	2.0	317.46	75.94
2.00×10^6	256	One	4	2.0	177.55	69.80
2.00×10^6	256	Two	0	2.0	51.23	45.29
2.00×10^6	256	Two	1	2.0	86.50	61.13
2.00×10^6	256	Two	2	2.0	128.36	89.07
3.88×10^6	512	One	4	2.0	219.50	107.15
3.88×10^6	512	One	6	2.0	149.23	130.29
3.88×10^6	512	Two	0	2.0	114.77	79.87
3.88×10^6	512	Two	1	2.0	120.50	85.72
3.88×10^6	512	Two	2	2.0	95.23	70.67

Table 7
Performance of the two-level method with respect to various values of Young's modulus *E*. The tests are run for the branching case with 2.00×10^6 unknowns, together with a coarse mesh with 6.60×10^4 unknowns.

np	E (g/(cm s^2))	Newton	fGMRES	Time (s)
256	6.0×10^6	2.0	51.23	45.29
256	1.4×10^7	2.0	68.46	55.75
256	6.0×10^7	2.0	119.23	67.13
512	6.0×10^6	2.0	86.73	27.68
512	1.4×10^7	2.0	105.23	33.20
512	6.0×10^7	2.0	186.46	56.11

Table 8
Performance of the two-level method with respect to various values of Poisson ratio. The tests are run for the branching case with 2.00×10^6 unknowns, together with a coarse mesh with 6.60×10^4 unknowns.

np	Poisson ratio	Newton	fGMRES	Time (s)
256	0.4	2.0	39.09	40.66
256	0.45	2.0	42.59	41.79
256	0.48	2.0	51.23	45.29
512	0.4	2.0	68.32	25.00
512	0.45	2.0	74.41	26.14
512	0.48	2.0	86.73	27.68

Table 9
Performance of the two-level method with respect to different combinations of fluid density ρ_f and structure density ρ_s. The tests are run for the branching case with a fine mesh with 2.00×10^6 unknowns and a coarse mesh with 6.60×10^4 unknowns on 512 processors.

ρ_f	ρ_s	Newton	fGMRES	Time (s)
0.01	1.0	2.0	304.41	109.51
0.1	1.0	2.4	169.09	57.64
1.0	1.2	2.0	86.73	27.68
10.0	1.0	2.0	115.14	31.14
100.0	1.0	3.0	209.82	65.68
1.0	0.01	2.0	134.73	36.65
1.0	0.1	2.0	122.82	34.67
1.0	10.0	2.0	73.00	25.83
1.0	100.0	2.0	74.00	32.11

5. Conclusion

In this paper, we studied a parallel domain decomposition algorithm with a two-level preconditioner for simulating blood flows

Please cite this article in press as: Wu Y, Cai X-C. A parallel two-level method for simulating blood flows in branching arteries with the resistive boundary condition. Comput Fluids (2011), doi:10.1016/j.compfluid.2010.11.015

Table 10
Performance of the two-level method with respect to various values of the resistance. The tests are run for the branching case with 2.00×10^6 unknowns, together with a coarse mesh with 6.60×10^4 unknowns.

np	Resistance (dyn·s/cm^5)	Newton	fGMRES	Time (s)
256	599.85	2.0	51.23	45.29
256	1000.0	2.0	55.96	46.83
256	2000.0	2.0	53.32	45.70
256	3000.0	2.0	52.68	45.35
512	599.85	2.0	86.73	27.68
512	1000.0	2.0	86.82	27.71
512	2000.0	2.0	93.36	28.61
512	3000.0	2.0	91.96	28.48

in the complaint arteries with the resistive outflow boundary condition. We demonstrated that the resistive boundary condition is an improvement over the zero-traction outflow boundary condition. We applied the resistive outflow boundary condition on two different artery models and we discussed the performance of a two-level preconditioner with the integral type resistive boundary condition. Based on a large number of numerical experiments, we are convinced that the two-level approach is effective and scalable with over one thousand processors.

Acknowledgments

Special thanks to Andrew Barker for his previous work on this project and to Zhenbi Su, Kendall Hunter, and Robin Shandas for helpful discussions and acquiring clinical data for our numerical experiments.

References

[1] Badia S, Quaini A, Quarteroni A. Splitting methods based on algebraic factorization for fluid–structure interaction. SIAM J Sci Comput 2008;30:1778–805.
[2] Balay S, Buschelman K, Eijkhout V, Gropp WD, Kaushik D, Knepley MG et al. PETSc users manual, technical report, Argonne National Laboratory; 2010.
[3] Barker AT, Cai X-C. Scalable parallel methods for monolithic coupling in fluid–structure interaction with application to blood flow modeling. J Comput Phys 2010;229:642–59.
[4] Barker AT, Cai X-C. Two-level Newton and hybrid Schwarz preconditioners for fluid–structure interaction. SIAM J Sci Comput 2010;32:2395–417.
[5] Bazilevs Y, Calo V, Zhang Y, Hughes T. Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. Comput Mech 2006;38:310–22.
[6] Bazilevs Y, Calo V, Hughes T, Zhang Y. Isogeometric fluid–structure interaction: theory, algorithms and computations. Comput Mech 2008;43:3–37.
[7] Cai X-C, Sarkis M. A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM J Sci Comput 1999;21:792–7.
[8] Cebral JR, Castro MA, Burgess JE, Pergolizzi RS, Sheridan MJ, Putman CM. Characterization of cerebral aneurysms for assessing risk of rupture by using patient-specific computational hemodynamics models. AJNR Am J Neuroradiol 2005;26:2550–9.
[9] Causin P, Gerbeau JF, Nobile F. Added-mass effect in the design of partitioned algorithms for fluid–structure problems. Comput Methods Appl Mech Eng 2005;194:4506–27.
[10] Dennis Jr JE, Schnabel RB. Numerical methods for unconstrained optimization and nonlinear equations. Philadelphia: SIAM; 1996.
[11] Eisenstat SC, Walker HF. Globally convergent inexact Newton method. SIAM J Optim 1994;4:393–422.
[12] Eisenstat SC, Walker HF. Choosing the forcing terms in an inexact Newton method. SIAM J Sci Comput 1996;17:16–32.
[13] Figueroa CA, Vignon-Clementel IE, Jansen KE, Hughes TJR, Taylor CA. A coupled momentum method for modeling blood flow in three-dimensional deformable arteries. Comput Methods Appl Mech Eng 2006;195:5685–706.
[14] Formaggia L, Gerbeau JF, Nobile F, Quarteroni A. On the coupling of 3D and 1D Navier–Stokes equations for flow problems in compliant vessels. Comput Methods Appl Mech Eng 2001;191:561–82.
[15] Karypis G, Aggarwal R, Schloegel K, Kumar V, Shekhar S. Metis/ParMETIS web page, University of Minnesota; 2010. <http://glaros.dtc.umn.edu/gkhome/views/metis>.
[16] Le Tallec P, Mouro J. Fluid structure interaction with large structural displacements. Comput Methods Appl Mech Eng 2001;190:3039–67.
[17] Michler C, van Brunmmelen EH, de Borst R. The relevance of conservation for stability and accuracy of numerical methods for fluid–structure interactions. Comput Methods Appl Mech Eng 2003;192:4195–215.
[18] Olufsen MS, Peskin CS, Kim WY, Pedersen EK, Nadim A, Larsen J. Numerical simulation and experimental validation of blood flow in arteries with structure-tree outflow conditions. Ann Biomed Eng 2000;28:1281–99.
[19] Owen SJ, Shepherd JF. Cubit Project Web Page, Sandia National Laboratories; 2010. <http://cubit.sandia.gov/>.
[20] Quarteroni A, Tuveri M, Veneziani A. Computational vascular fluid dynamics: problems, models and methods. Comput Visual Sci 2000;2:163–97.
[21] Saad Y, Schultz MH. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear system. SIAM J Sci Stat Comput 1986;7:856–69.
[22] Saad Y. Iterative methods for sparse linear systems. SIAM; 2003.
[23] Taylor CA, Draney MT. Experimental and computational methods in cardiovascular fluid mechanics. Ann Rev Fluid Mech 2004;36:197–231.
[24] Vignon IE, Taylor CA. Outflow boundary conditions for one-dimensional finite element modeling of blood flow and pressure waves in arteries. Wave Motion 2004;39:361–74.
[25] Vignon-Clementel IE, Figueroa CA, Jansen KE, Taylor CA. Outflow boundary conditions for three-dimensional finite element modeling of blood flow and pressure in arteries. Comput Methods Appl Mech Eng 2006;195:3776–96.
[26] Tezduyar TE, Behr M, Liou J. A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space–time procedure: I. The concept and the preliminary numerical tests. Comput Methods Appl Mech Eng 1992;94:339–51.