Simulation of Branching Blood Flows on Parallel Computers[1]

Xue Yue, Feng-Nan Hwang, Robin Shandas, Xiao-Chuan Cai

(XY/XCC): Dept. of Computer Science, Univ. of Colorado at Boulder, Boulder, CO 80309
(FNH): Dept. of Applied Mathematics, Univ. of Colorado at Boulder, Boulder, CO 80309
(RS): Dept. of Pediatrics, Health Sciences Center, Univ.of Colorado, Denver, CO 80218

Abstract

We present a fully parallel nonlinearly implicit algorithm for the numerical simulation of some branching blood flow problems, which require efficient and robust solver technologies in order to handle the high nonlinearity and the complex geometry. Parallel processing is necessary because of the large number of mesh points needed to accurately discretize the system of differential equations. In this paper we introduce a parallel Newton-Krylov-Schwarz based implicit method, and software for distributed memory parallel computers, for solving the nonlinear algebraic systems arising from a Q2-Q1 finite element discretization of the incompressible Navier-Stokes equations that we use to model the blood flow in the left anterior descending coronary artery.

Key words: Blood flow modeling, incompressible Navier-Stokes equations, implicit methods, domain decomposition methods, parallel computing

1. Introduction

In recent years there is growing interest in parallel computational techniques for the study of blood flow dynamics in vascular systems. Numerous computational models have been developed to describe the local blood flow field and to simulate the response of the vessel walls under certain hemodynamic conditions. The blood characteristics of interests are often abnormal local flow structures such as vortices, flow separations, recirculation and stagnation areas; blood pressure drop; high shear stress regions and system total energy dissipation[4, 5]. These characteristics play critical roles in the formation of serious vascular diseases such as atherosclerotic lesions and thrombus formation. The goal of the computational biofluid research is to provide reliable tools to illustrate the hemodynamic properties under specific conditions and to predict their changes when certain disturbances arise. Good simulation results can be used clinically to help physicians to understand vascular diseases. To resolve the detailed changes of the hemodynamic equations in complex and delicate vascular systems the use of highly refined unstructured finite element meshes is essential. This implies that the computational time would be increased dramatically when CPU computer is used. The large amount of required computational time makes some simulations infeasible. Consequently, a fast parallel computational algorithm and unstructured mesh generation techniques are needed to overcome such limitations.

The purpose of this paper is to develop a numerical software system for the simulation of complex biofluid problems. A parallel Newton-Krylov-Schwarz(NKS) based nonlinearly implicit algorithm is presented to solve the unsteady incompressible Navier-Stokes equations. An ANSYS and C++ based

[1]The work was partially supported by the National Science Foundation, CCR-0219190, ACI-0072089 and ACI-0305666.

unstructured mesh generation technique is introduced to discretize the multi-branched numerical models. The numerical examples presented here demonstrated the good scalability and the robustness of the proposed algorithm.

The rest of this paper is organized as follows. In the next section, we briefly mention the incompressible Navier-Stokes equations, followed by a description of a Newton-Krylov nonlinear solver and a parallel Schwarz preconditioner for the saddle-point type Jacobian system. In Section 3, numerical results on a branching flow problem are presented. Concluding remarks are given in Section 4.

2. Blood Flow Model and Parallel Solution Algorithm

Under some certain assumptions, blood flows in large arteries can be described by the unsteady incompressible Navier-Stokes equations [6]:

$$\begin{cases} \rho\left(\dfrac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) - \mu\Delta\boldsymbol{u} + \nabla p = 0 & \text{in } \Omega\times(0,T), \\ \nabla\cdot\boldsymbol{u} = 0 & \text{in } \Omega\times(0,T), \\ \boldsymbol{u} = 0 & \text{on } \Gamma_{wall}\times(0,T), \\ \boldsymbol{u} = g & \text{on } \Gamma_{in}\times(0,T), \\ -p\boldsymbol{n} + \mu\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} = 0 & \text{on } \Gamma_{out}\times(0,T), \\ \boldsymbol{u} = \boldsymbol{u}_0 & \text{in } \Omega \text{ at } t=0, \end{cases} \tag{1}$$

where $\boldsymbol{u} = (u_1, u_2)^T$ is the velocity, p is the pressure, ρ is the fluid density, and μ is the viscosity. Here, we impose a periodic velocity profile on the inflow boundary, Γ_{in}, an artificial free condition on the outflow boundary, Γ_{out}, and the no-slip condition on the wall, Γ_{wall}. For $\boldsymbol{u}_0$, we assume that the flow is steady-state at the beginning of computation. To discretize (1), we use an implicit backward Euler finite difference method in the time domain and a Q2-Q1 mixed finite element method in the spatial domain on a given unstructured quadrilateral mesh, $\mathcal{T}_h = \{K\}$. Let V_h and P_h be a pair of finite element spaces for the velocity and pressure: $V_h = \{\boldsymbol{v} \in (C^0(\Omega)\cap H^1(\Omega))^2 : \ \boldsymbol{v}|_K \in Q_2(K)^2, \ K\in\mathcal{T}_h\}$ and $P_h = \{p \in C^0(\Omega)\cap L^2(\Omega) : \ p|_K \in Q_1(K), \ K\in\mathcal{T}_h\}$. The weighting and trial velocity function spaces V_h^0 and V_h^g are $V_h^0 = \{\boldsymbol{v}\in V_h : \boldsymbol{v} = 0 \text{ on } \Gamma_{in}\cup\Gamma_{wall}\}$ and $V_h^g = \{\boldsymbol{v}(\cdot,t)\in V_h, t\in[0,T] : \ \boldsymbol{v} = g \text{ on } \Gamma_{in} \text{ and } \boldsymbol{v} = 0 \text{ on } \Gamma_{wall}\}$. Similarly, P_h is used for both the weighting and trial pressure function spaces. A fully nonlinear implicit mixed finite element method takes the form: Find $\boldsymbol{u}_h^{n+1}\in V_h^g$ and $p_h^{n+1}\in P_h$, such that

$$B(\boldsymbol{u}_h^{n+1}, p_h^{n+1}; \boldsymbol{v}, q) = 0 \qquad\qquad \forall(\boldsymbol{v},q)\in V_h^0\times P_h \tag{2}$$

with

$$B(\boldsymbol{u},p;\boldsymbol{v},q) = \left(\rho\left(\frac{\boldsymbol{u}-\boldsymbol{u}^n}{\Delta t}\right),\boldsymbol{v}\right) + ((\rho\nabla\boldsymbol{u})\cdot\boldsymbol{u},\boldsymbol{v}) + (\mu\nabla\boldsymbol{u},\nabla\boldsymbol{v}) - (\nabla\cdot\boldsymbol{v},p) - (\nabla\cdot\boldsymbol{u},q),$$

where $\boldsymbol{u}^n$ is the velocity at the current time step, and $\boldsymbol{u}^{n+1}$ and p^{n+1} are unknown velocity and pressure, respectively, at the next time step. Or, equivalently, at each time step, one needs to solve a large, sparse, nonlinear algebraic system $F(x) = 0$, where the vector x corresponds to the nodal values of $\boldsymbol{u}_h = (u_h^1, u_h^2)$ and p_h at time $t = (n+1)\Delta t$. In this paper, we employ a Newton-Krylov-Schwarz algorithm(NKS [2]) to solve the nonlinear system. NKS has three main components: an inexact Newton with backtracking method (INB) as the nonlinear solver, a Krylov subspace method as the linear solver, and a Schwarz-type

method as the preconditioner. INB can be described as follows: Let $x^{(0)}$ be any given initial guess, and $x^{(k)}$ be the current approximate solution. Then a new approximate solution $x^{(k+1)}$ can be computed by the following steps: First find the inexact Newton direction $s^{(k)}$ such that

$$||F(x^{(k)}) - J(x^{(k)})s^{(k)}||_2 \leq \eta_k ||F(x^{(k)})||_2.$$

Then compute the new approximate solution $x^{(k+1)} = x^{(k)} - \lambda^{(k)} s^{(k)}$, by selecting a step length λ using line search to satisfies $f(x^{(k)} - \lambda^{(k)} s^{(k)}) \leq f(x^{(k)}) - \alpha \lambda^{(k)} \nabla f(x^{(k)})^T s^{(k)}$, where the merit function f is defined as $||F(x)||_2^2/2$. The parameter α is chosen to be 10^{-4} throughout the computation. In INB, the scalar η_k is often called the "forcing term", which is used to determine how accurately the Jacobian system needs to be solved by some iterative methods, such as a Krylov subspace type method GMRES. In the paper, we compute $s^{(k)}$ by approximately solving a preconditioned linear system

$$M^{-1} J s^{(k)} = M^{-1} F(x^{(k)}),$$

where the Jacobian J is a large, sparse, nonsymmetric, and indefinite matrix. Our preconditioner is an extension of the one-level additive Schwarz preconditioner [7] defined as follows. Let $\Omega_i, i = 1, ..., N$ be an non-overlapping partition of Ω, and Ω_i' an overlapping extension of Ω_i with the boundary $\partial\Omega_i'$. Here N is the number of processors of the parallel computer. We define the associated subdomain velocity space as $V_h^i = V_h^0 \cap (H_0^1(\Omega_i'))^2$, and the associated subdomain pressure space as $P_h^i = \{q_h \in L^2(\Omega_i') : q_h = 0 \text{ on } \partial\Omega_i'/\partial\Omega\}$.

Let $R_i : V_h^g \times P_h \rightarrow V_h^i \times P_h^i$ be a global-to-local restriction operator associated with Ω_i', and R_i returns all degrees of freedom (both velocity and pressure) associated with the subspaces $V_h^i \times P_h^i$. Then, the local-to-global interpolation operator R_i^T can be defined as the transpose of R_i. The multiplication of R_i and R_i^T with a vector does not involve any arithmetic operation, but does involve communication in a distributed parallel implementation. The restriction operator R_i collects the data from neighboring subdomains, and the prolongation operator R_i^T sends a partial solution to neighboring subdomains. Using the restriction matrix, we write the one-level additive Schwarz preconditioner in the matrix form as $M^{-1} = \sum_{i=1}^{N} R_i^T J_i^{-1} R_i$, where $J_i = R_i J R_i^T$.

3. Numerical Experiments

We apply our parallel NKS technique to the simulation of pulsatile blood flows in the left anterior descending coronary artery. Our main focus is the effect of the inlet velocity fluctuation on the flow hemodynamic properties. The flow shear stress and local flow structures such as stagnation, recirculation, separation and reversal regions are investigated. Our parallel simulation system consists of several elements including (1) An ANSYS and C++ based finite element mesh generator; (2) A mesh partitioner implemented via Parmetis[3]; (3) A scalable parallel linear and nonlinear solver implemented through PETSc[1]; and (4) A Tecplot based scientific visualization system.

Figure 1 (A) shows the model geometry provided by the Cardiovascular Flow Research Laboratory at The Children's Hospital of University of Colorado Health Science Center. It has one inlet with diameter of 4.62mm and a total of 8 outlets. The minimum outlet diameter is 1.65mm. The model is discretized with 17783 quadrilateral finite elements with 73281 mesh points. Two cases with different inlet boundary velocity are simulated. Figure 1 (B) shows the time history of the input velocity at the center of the inlet. The inlet flow mass rate for both cases are set to be positive during the whole simulation time. The analysis was performed for two cardiac cycles with time step size $\Delta t = 0.05(sec)$.

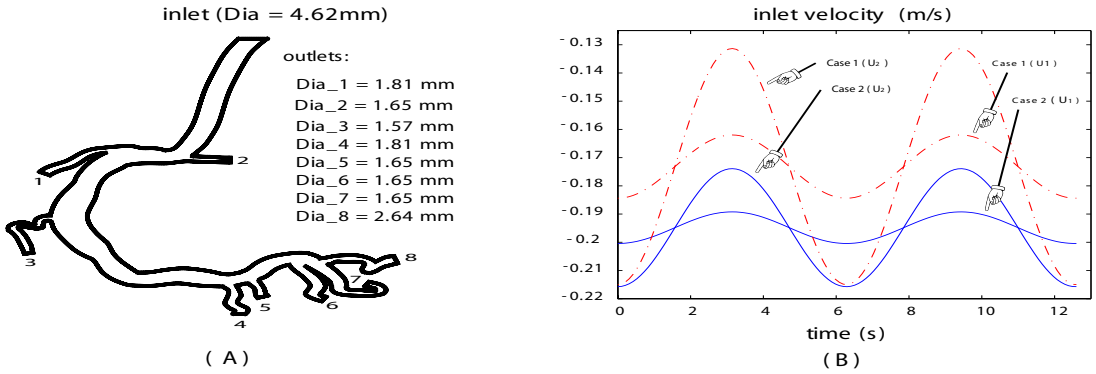

Figure 1: (A) Model Geometry (B) Inlet Velocity

Figure 2 describes the local flow structure evolvement along time for Case 1 and cCase 2. It is shown that Case 1 has larger recirculation and stagnation regions at most time. The local flow structure in Case 1 evolves more dramatically than Case 2. For example, the flow going through outlet 1 in Case 1 decreases very fast with time. When time approaches to 3.78 second, reverse flow occurs. This is very different from Case 2, which shows no reverse flow during the entire simulation. The shear stress at every mesh point is computed using $\tau_{ij} = -p\delta_{ij} + \mu\,(\partial u_i/\partial x_j + \partial u_j/\partial x_i), i, j = 1, 2$, where δ_{ij} is the Kronecker delta. The maximum shear stress at each mesh point is derived as: $\tau_{max} = (((\tau_{11} - \tau_{22})/2)^2 + \tau_{12}^2)^{1/2}$. Table 1 lists the maximum shear stress at different time for the two cases. The maximum shear stress for both cases occurs in the same element, which is on the wall in the confluence area of outlet 2 to the main vessel. Case 2 has higher shear stress than Case 1 at all time instants. One explanation is that Case 2 has a higher transverse velocity component close to the vessel wall where the recirculation area is smaller than that of Case 1.

Our parallel software is developed using the Portable, Extensible Toolkit for Scientific Computing (PETSc) library [1], from Argonne National Laboratory and all tests are performed on a cluster of distributed memory Linux PCs, with MPI as the communication language. Our main concern is the scalability of the algorithm in terms of the total computing time which includes both the CPU time and the communication time. The number of the processors and the corresponding computing time required to reach the converged solution for both cases are listed in Table 2. In both cases, the computing time drops dramatically when the number of processors is increased. A closer look shows that if we increase the number of processors by a factor of 2, the computing time is cut by a factor close to 1/2 which demonstrates the near ideal linear scalability of the proposed nonlinear parallel algorithm and our parallel implementation.

4. Concluding Remarks

In this paper, a fully parallel finite element algorithm and its corresponding distributed memory software implementation are presented for the numerical simulation of branching blood flows. The

τ_{max}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
Case 1	5.0815	4.8061	4.2693	3.7069	3.2770	3.1623	3.4172	3.9313	4.4889	4.9358
Case 2	5.1092	4.9937	4.7472	4.4637	4.2642	4.2172	4.3218	4.5606	4.8443	5.0535

Table 1: Comparison of maximum shear stress(N/m^2)

number of processors	1	2	4	8	16
computing time (Case 1)	8124.3	4793.5	2213.4	1326.7	886.2
computing time (Case 2)	7868.2	4643.4	1982.7	1283.5	842.1

Table 2: The number of processors vs. the total computing time in second

unsteady incompressible Navier-Stokes equations are solved using a parallel Newton-Krylov-Schwarz based nonlinearly implicit domain decomposition method. It was found that the inlet velocity fluctuation plays crucial role in determining the flow hemodynamic properties. The numerical experiments also verified the scalability and robustness of the nonlinear parallel algorithm developed in this paper. It showed a large reduction of computational time when we increase the number of processors, which reveals one of the potentials of the proposed computational framework for solving large scale simulation problems.

References

[1] S. Balay, K. Buschelman, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, B. Smith, H. Zhang, PETSc Users manual, ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2003.

[2] X.-C. Cai, W. D. Gropp, D. E. Keyes, R. G. Melvin, and D. P. Young, Parallel Newton-Krylov-Schwarz algorithms for the transonic full potential equation, SIAM J. Sci. Comput. 19 (1998) 246-265.

[3] G. Karypis, R. Aggarwal, K. Schloegel, V. Kumar, and S. Shekhar, METIS home page, http://www-users.cs.umn.edu/ karypis/metis/.

[4] Y. Khunatorn, S. Mahalingam, C. DeGroff, and R. Shandas, A fluid dynamic study of the total cavopulmonary connection in the Fontan operation, Proceedings of the 11th International Conference on Mechanics in Medicine and Biology, April, 2000.

[5] Y. Khunatorn, C. DeGroff, S. Mahalingam, R. Shandas, Influence of connection geometry and SVC-IVC flow rate ratio on flow structures within the total cavopulmonary connection: A numerical study, J. Biomechanical Engineering, 2002.

[6] A. Quarteroni, M. Tuveri, and A. Veneziani, Computational vascular fluid dynamics: problems, models and methods, Comp. and Visual. Sci., 2 (2000) 163-197.

[7] B. Smith, P. Bjorstad, and W. Gropp, Domain Decompostion: Parallel Multilevel Methods for Ellipic Partial Differential Equations, Cambridge University Press, Combridge, 1996.

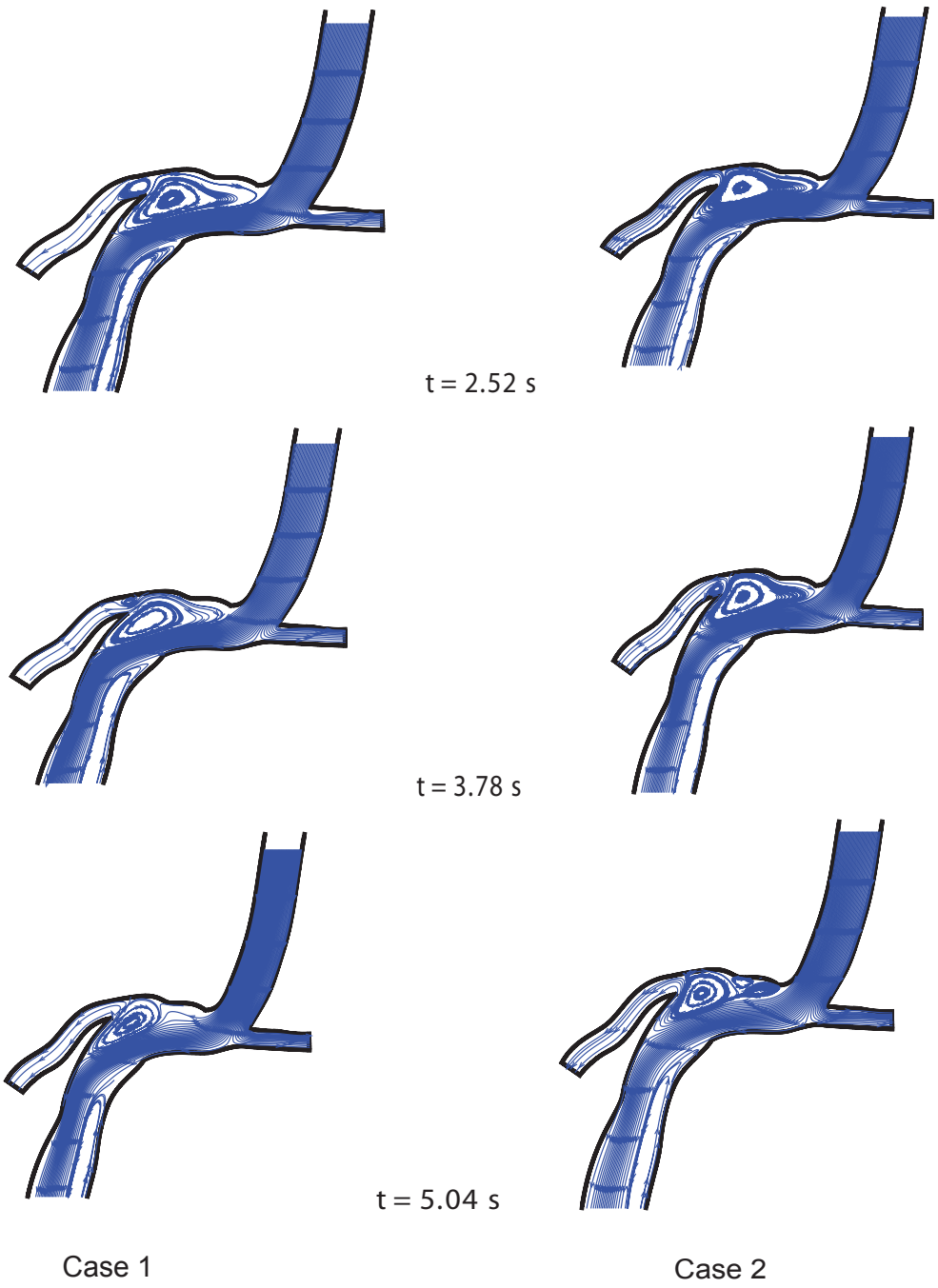

Figure 2: Streamline contour plots