

**Parallel Domain Decomposition Methods For Simulating  
Blood Flows In Three-Dimensional Compliant Arteries**

by

**Yuqi Wu**

B.S., Shantou University, 2004

M.S., University of Houston, 2006

A thesis submitted to the  
Faculty of the Graduate School of the  
University of Colorado in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
Department of Applied Mathematics

2012

This thesis entitled:  
Parallel Domain Decomposition Methods For Simulating Blood Flows In Three-Dimensional  
Compliant Arteries  
written by Yuqi Wu  
has been approved for the Department of Applied Mathematics

---

Xiao-Chuan Cai

---

Tom Manteuffel

---

Kendall Hunter

---

Congming Li

---

Steve McCormick

Date \_\_\_\_\_

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Wu, Yuqi (Ph.D., Applied Mathematics)

Parallel Domain Decomposition Methods For Simulating Blood Flows In Three-Dimensional Compliant Arteries

Thesis directed by Prof. Xiao-Chuan Cai

Numerical simulation of blood flows in compliant arteries is becoming an useful tool in studying the sophisticated hemodynamics in the human circulation system. Accurate modeling is important in prediction and treatment of artery diseases. In this thesis, we propose and study a parallel domain decomposition method for solving the corresponding fluid-structure interaction problem in three-dimensional space, with emphasis on the strong coupling between fluid and structure and on the parallel scalability of the solution algorithm.

We model the fluid-structure interaction by using a monolithically coupled system of linear elasticity equations for the arterial walls and incompressible Navier-Stokes equations for the blood. The fluid equations are derived in an arbitrary Lagrangian-Eulerian framework to address the complicated moving boundaries and keep track of the coupling on the interface. A finite element method based on the unstructured mesh is introduced and validated for discretizing the problem in space, and a fully implicit scheme is used for the temporal discretization.

For solving the nonlinear systems arising from the fully coupled discretization, we develop a class of Newton-Krylov-Schwarz algorithms. The investigation focuses on the parallel efficiency of the fully implicit solution algorithm, as well as the performance of one-level and two-level additive Schwarz preconditioners used in accelerating the convergence of the Newton-Krylov algorithm. Simulations based on some patient-specific pulmonary artery geometries are performed on a large scale supercomputer. Our algorithm is shown to have excellent parallel scalability with over three thousand processors and for problems with millions of unknowns, and is also robust with respect to several important physical parameters including the fluid density, the structure density, the Reynolds number, and the Poisson ratio.

## Dedication

To my wife, my parents, and my grandma.

## Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Xiao-Chuan Cai, who has introduced me to the fascinating world of scientific computing, and has given me the opportunity to work on this very interesting research. I am particularly grateful to him for the confidence he has always had in me and for the enthusiasm in scientific research he has passed on to me. For years, Professor Cai has given me so much of his time in guiding my research and has always encouraged me to further investigate aspects of a problem. Without his advice, support and patience, I would not have completed this thesis.

I would like to thank Professors Kendall Hunter, Tom Manteuffel, Steve McCormick, and Congming Li for serving on my committee, reviewing my work, and providing helpful feedback and advice for this research. I would also like to thank Dr. Andrew Barker and Dr. Feng-Nan Hwang for their senior advice and previous work on this research. Thank you to Craig Lanning and Mark Reusser at Department of Bioengineering for all the helpful discussions and acquisition clinical data for numerical experiments.

I am indebted to my wonderful colleagues and friends, Doctors Rongliang Chen, Si Liu, Lei Tang, Chao Yang, and Ms. Cui Cong for their help and willingness to listen to and discuss my work. I would also like to acknowledge the PETSc team of Argonne National Laboratory for their software support and many fruitful suggestions.

Last, but not the least, I would like to express my heartfelt appreciation to my family, especially my wife Fang Liang, for their love and encouragement over years.

## Contents

<b>Chapter</b>	
<b>1</b>	<b>Introduction</b> . . . . . 1
1.1	Overview . . . . . 1
1.2	Modeling fluid-structure interaction problem . . . . . 2
1.2.1	Physics of blood flow and artery wall . . . . . 3
1.2.2	Simulating fluid in moving domain . . . . . 3
1.2.3	Coupling formulations for fluid-structure interaction . . . . . 5
1.2.4	Boundary conditions . . . . . 5
1.3	Solution algorithms . . . . . 7
1.4	Outline of the thesis . . . . . 9
<b>2</b>	<b>Mathematical models and discretizations</b> . . . . . 11
2.1	Governing equations—strong form . . . . . 11
2.1.1	Structure model . . . . . 13
2.1.2	Moving domain description . . . . . 14
2.1.3	Fluid model . . . . . 15
2.1.4	Coupling conditions . . . . . 16
2.2	Governing equations—weak form . . . . . 17
2.3	Spatial discretization . . . . . 20
2.3.1	Stabilized finite element method . . . . . 20

2.3.2	Semi-discrete fully coupled problem . . . . .	22
2.4	Fully implicit time discretization . . . . .	27
<b>3</b>	<b>A monolithic nonlinear solver for the coupled system of equations</b>	<b>31</b>
3.1	Overview of Newton-Krylov-Schwarz methods . . . . .	31
3.2	Evaluation of Jacobian . . . . .	32
3.3	One-level restricted additive Schwarz preconditioner . . . . .	33
3.3.1	Subdomain solver . . . . .	35
3.4	Some theory of Schwarz methods . . . . .	36
3.5	Two-level Schwarz preconditioner . . . . .	37
3.5.1	Interpolation and restriction . . . . .	37
3.5.2	Parallel implementation . . . . .	38
3.5.3	Coarse-level solver . . . . .	40
<b>4</b>	<b>2D FSI simulations with resistance boundary condition</b>	<b>42</b>
4.1	Impact of different outflow boundary conditions . . . . .	43
4.1.1	2D straight tube . . . . .	43
4.1.2	2D bifurcating artery . . . . .	44
4.2	Performance and parallel scalability . . . . .	52
<b>5</b>	<b>3D FSI Simulations of blood flows in compliant arteries</b>	<b>59</b>
5.1	Validation . . . . .	59
5.1.1	Structure solver . . . . .	60
5.1.2	Fluid solver in moving domain . . . . .	60
5.2	Benchmark test case . . . . .	63
5.3	Two-branch artery case . . . . .	67
5.4	Complex branching artery case . . . . .	70
5.5	Robustness to physical parameters . . . . .	72

5.6	Two-level results . . . . .	75
5.6.1	Benchmark test case . . . . .	76
5.6.2	Two-branch artery case . . . . .	78
5.6.3	Parameter selection . . . . .	78
<b>6</b>	<b>Conclusions and future work</b>	<b>85</b>
6.1	Conclusions . . . . .	85
6.2	Future work . . . . .	86
	<b>Bibliography</b>	<b>88</b>

## Tables

### Table

4.1	Performance of the one-level preconditioner with the zero-traction and resistance outflow boundary conditions. The tests are carried on the 2D straight tube problem, associated with the fine mesh of $2.01 \cdot 10^6$ degrees of freedom. “ $np$ ” denotes the number of processors. “Newton” denotes the average Newton iterations per time step. “GMRES” denotes the average GMRES iterations per Newton step. “Time” refers the average compute time, in seconds, per time step. . . . .	52
4.2	Performance of the cascade two-level preconditioner with the zero-traction and resistance outflow boundary conditions. The tests are carried on the 2D straight tube problem, associated with the fine mesh of $2.01 \cdot 10^6$ degrees of freedom and the coarse mesh of $1.30 \cdot 10^5$ 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. . . . .	53
4.3	Performance of two different preconditioners with increasing number of subdomains for the 2D bifurcating artery problem. The tests are carried on a mesh with $2.00 \cdot 10^6$ unknowns, using the resistance outflow boundary condition. . . . .	54

4.4	The effect of various choices of the preconditioners for the 2D 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. . . . .	56
4.5	The effect of overlapping parameter for the one-level and two-level preconditioners for the 2D bifurcating artery problem with the resistance outflow boundary condition.	57
5.1	Convergence to the analytic solutions at different phases with respect to the discretization size $h$ for the 3D fluid problem with moving domain. . . . .	63
5.2	Performance with respect to the number of processors for different subdomain solvers for the 3D benchmark problem. The tests are carried on a mesh with $1.25 \cdot 10^6$ unknowns with a fixed overlapping size $\delta = 1$ . “ $np$ ” denotes the number of processors. “NI” denotes the average number of Newton iterations per time step. “GMRES” denotes the average number of GMRES iterations per Newton step. “time” refers to the average compute time, in seconds, per time step. “fill-in” refers to the average fill-in ratio needed in the subdomain factorization per iteration. . . . .	66
5.3	Performance with respect to the number of processors for different subdomain solvers for the two-branch model. The tests are carried on a mesh with $1.24 \cdot 10^6$ unknowns with a fixed overlapping size $\delta = 1$ . “ $np$ ” denotes the number of processors. “NI” denotes the average number of Newton iterations per time step. “GMRES” denotes the average number of GMRES iterations per Newton step. “time” refers to the average compute time, in seconds, per time step. “fill-in” refers to the average fill-in needed in the factorization per iteration. . . . .	71

5.4	The effect of various choices of overlapping parameter $\delta$ on different mesh sizes and number of processors. These tests are for the complex branching problem. “ $np$ ” denotes the number of processors. “Newton” denotes the average Newton iteration per time step. “GMRES” denotes the average GMRES iterations per Newton step. “time” refers to the average compute time, in seconds, per time step. . . . .	74
5.5	Performance for different combinations of fluid density $\rho_f$ and wall structure density $\rho_s$ for the complex branching model. The dynamic viscosity $\nu_f$ is kept as $0.035 \text{ cm}^2/s$ . The tests are run on a mesh with $1.83 \cdot 10^6$ degrees of freedom and 1024 processors. . . . .	75
5.6	Performance for various values of Young’s modulus $E_s$ and Poisson ratio $\nu$ for the complex branching model. The tests are run on a mesh with $1.83 \cdot 10^6$ degrees of freedom and 1024 processors. . . . .	76
5.7	Performance of the one-level and two-level preconditioners with respect to the increasing number of subdomains for the 3D benchmark problem. The heading “unknowns” reports the problem size, “ $np$ ” denotes the number of processors, “Newton” denotes the average Newton iteration per time step, “GMRES” denotes the average GMRES iterations per Newton step, “time” refers to the average compute time, in seconds, per time step. . . . .	77
5.8	Performance of the one-level and two-level preconditioners with respect to the increasing number of subdomains for the two-branch artery problem. . . . .	79
5.9	Effect of different coarse mesh sizes on the two-level preconditioner. These results are for the 3D benchmark problem. The heading “coarse size” represents the number of unknowns on the coarse mesh. . . . .	81
5.10	Effect of different coarse mesh sizes on the two-level preconditioner. These results are for the two-branch artery problem. The heading “coarse size” represents the number of unknowns on the coarse mesh. . . . .	81

- 5.11 Performance of the two-level preconditioner with various parameter values of coarse solver tolerance. These results are for the 3D benchmark problem. The heading “coarse rtol” refers to the relative tolerance for the coarse solve, and “coarse its” denotes the average number of linear iterations spent in the coarse-level solver per fine fGMRES iteration. . . . . 82
- 5.12 Performance of the two-level preconditioner with various parameter values of coarse solver tolerance. These results are for the two-branch artery problem. The heading “coarse rtol” refers to the relative tolerance for the coarse solve, and “coarse its” denotes the average number of linear iterations spent in the coarse-level solver per fine fGMRES iteration. . . . . 83
- 5.13 Performance of the two-level preconditioner with respect to different fine-level subdomain solvers. These results are for the 3D benchmark problem. The headings “fine sub-solver” denotes the choices of subdomain solver for the fine-level preconditioner, where LU stands for the standard LU factorization, BLU for the point-block LU factorization, and BILU(1) for the point-block ILU with one level of fill-ins. . . . 83

## Figures

### Figure

- 2.1  $\Omega_s^0$  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,  $\Gamma_i$  and  $\Gamma_o$ , respectively, are fixed and the boundaries  $\Gamma_s$  of the structure domain are also fixed at the inlets and outlets.  $\Gamma_w^t$  represents the fluid-structure interface in current configuration. . . . 12
- 2.2 Example finite element mesh for the fluid-structure interaction problem (top). The mesh is generated so that the interface between the fluid and structure is aligned with the mesh. The elements of the fluid (bottom left) and the elements of the structure (bottom right) conform on the interface. . . . . 23
- 3.1 Sample partition of the domain into 4 subdomains by using ParMETIS. The fluid elements and structure elements are marked with different colors. The top figure shows the partition into non-overlapping subdomains, and the bottom represents a corresponding partition into overlapping subdomains with  $\delta = 2$ . The shaded elements in blue and green represent the corresponding fluid elements and structure elements extended from the non-overlapping subdomains. . . . . 34

3.2	Example partitions of the fine mesh and the coarse mesh of a pulmonary artery by using ParMETIS. The fluid element and structure elements are marked with different colors. The left figure shows the partition of the fine mesh, and the right represents a corresponding partition of the coarse mesh. The elements of a coarse subdomain are assigned to the corresponding processor that contains the fine mesh elements from the same subdomain. . . . .	39
4.1	The setup of the 2D straight tube problem and the inlet flow rate from clinical data with a polynomial fitting. . . . .	43
4.2	Flow waves at inlet and outlet, wall pressure and displacement over one cardiac cycles, obtained using resistance and zero-traction outlet boundary conditions for the 2D straight tube problem. Top left figure represents the inlet and outlet flow rate, and top right figure shows the wall displacement and pressure using the resistance boundary condition, while the bottom figures represent the results using the zero-traction outflow boundary condition. . . . .	45
4.3	Streamlines at the peak systole and mid-diastole phases for the 2D straight tube problem using the resistance outflow boundary condition. The artery walls are colored by the magnitude of the displacement and fluid streamlines are colored by the velocity in the X direction. The top figure represents the peak systole phase, the bottom figure for the mid-diastole phase. . . . .	46
4.4	Streamlines at the peak systole and mid-diastole phases for the 2D straight tube problem using the zero-traction outflow boundary condition. The artery walls are colored by the magnitude of the displacement and fluid streamlines are colored by the velocity in the X direction. The top figure represents the peak systole phase, the bottom figure for the mid-diastole phase. . . . .	46
4.5	Geometric details of the bifurcating artery in 2D. A 75% area reduction stenosis is shown on one of the branches. . . . .	47

- 4.6 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, 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 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 using zero-traction boundary conditions are shown in the bottom plots. . . . . 48
- 4.7 Fluid velocity magnitude and pressure at peak systole for the 2D bifurcating artery using 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. . . . 50
- 4.8 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. . . . . 51

4.9	Example partitions of the fine mesh and the coarse mesh of a 2D straight tube 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. Note that the outlet boundary on the right is shared by two different subdomains. . . . .	53
4.10	Speedup versus number of processors for the 2D straight tube problem with the resistance outflow boundary condition. Results for the problem with 2.01 million unknowns are on the left, and results for the problem with 3.99 million unknowns show on the right. . . . .	55
4.11	Speedup versus number of processors for the 2D bifurcating artery model with the resistance outflow boundary condition. Results for the problem with 2.00 million unknowns is on the left, and results for the problem with 3.88 million unknowns show on the right. . . . .	55
4.12	Weak scaling of the algorithm for the 2D 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: $2.01 \cdot 10^6$ for 256 processors and $7.99 \cdot 10^6$ for 1024 processors. . . . .	55
5.1	Convergence to the analytic solution (at $t = 0.5s$ ) with respect to the spatial discretization size $h$ for the 3D structure test problem. The y-axis reports $L^2$ norm of the displacement error. The temporal discretization is implemented by the backward Euler scheme with the time step size of $\Delta t = 0.0025s$ . . . . .	61

5.2	Comparisons of the computed solutions with the exact analytic solutions of the 3D fluid problem with moving domain. The tests are carried on a mesh with $2.17 \cdot 10^6$ unknowns. The left plot shows the flow rate at the inlet and outlet boundary during one cycle, comparing to the exact analytic flow rate. The right plot shows the computed pressure at the center of the cylinder during one cycle, comparing to the exact analytic pressure at the center. . . . .	62
5.3	Pressure wave propagation (top row) and structure deformation (bottom row) for the 3D straight cylinder case. The simulation is run on a mesh with $2.41 \cdot 10^6$ elements and $3.08 \cdot 10^6$ degrees of freedom. The deformation is amplified by a factor of 12 for visualization purpose only. . . . .	65
5.4	Parallel speedup, average compute time per time step, and average number of GMRES iterations per Newton iteration for the 3D benchmark problem with increasing number of processors. The number of unknowns of each problem is listed in the legend. The tests are run with a fixed overlapping size $\delta = 1$ and BILU(1) is the subdomain solver. . . . .	65
5.5	Flow and pressure at the inlet and outlets over one cardiac cycle for the two-branch artery model, obtained using the resistance outflow boundary condition. Figure on the left represents the flow rate at the inlet and outlets, and figure on the right shows the fluid pressure at the inlet and outlets. The test is carried on a mesh with $3.57 \cdot 10^6$ element and $4.61 \cdot 10^6$ degrees of freedom. . . . .	68
5.6	Flow in part of the pulmonary artery with two branches at the peak systole (left) and the early diastole (right). The fluid streamlines are colored by velocity magnitude. . . . .	68
5.7	Arterial wall velocity vectors obtained at two points of the cardiac cycle: peak systole (left) and early diastole (right). . . . .	69

- 5.8 Parallel speedup, average compute time per time step, and average number of GMRES iterations per Newton iteration for the two-branch problem with increasing number of processors. The number of unknowns of each mesh is listed in the legend. The tests are run with a fixed overlapping size  $\delta = 1$ . . . . . 70
- 5.9 Results of the simulation of part of the pulmonary artery. In the large images, the fluid shaded by pressure is shown on the left and the fluid velocity colored in its magnitude is shown on the right. The artery wall is shown in a solid shade in both images. In the inset images, the artery wall shaded by the norm of the displacement is shown on the left and the fluid streamlines colored by vorticity is shown on the right. . . . . 73
- 5.10 Parallel speedup, average compute time per time step, and average GMRES iterations per Newton iteration for the complex branching problem with increasing number of processors. The number of unknowns of each problem is listed in the legend. The tests are run with a fixed overlapping size  $\delta = 1$ . . . . . 73
- 5.11 Weak scaling of the one-level and two-level preconditioners for the 3D benchmark problem. The reported time and linear iterations are the average compute time per time step and the average GMRES iterations per Newton step, respectively. In these tests, the number of unknowns increases with the number of processors:  $7.18 \cdot 10^5$  for 480 processors,  $1.25 \cdot 10^6$  for 930 processors, and  $3.08 \cdot 10^6$  for 2048 processors. . . . . 79
- 5.12 The linear iteration counts for the coarse-level solver with respect to the increasing number of subdomains for the two-branch artery problem. This plot shows the average number of linear iterations spent on the coarse-level per fine fGMRES iteration. The tests are carried on a mesh with  $1.24 \cdot 10^6$  unknowns. . . . . 80
- 5.13 Weak scaling of the one-level and two-level preconditioners for the two-branch artery problem. In these tests, the number of unknowns increases with the number of processors:  $3.33 \cdot 10^5$  for 221 processors,  $6.33 \cdot 10^5$  for 421 processors,  $1.24 \cdot 10^6$  for 833 processors, and  $4.61 \cdot 10^6$  for 3072 processors. . . . . 80

# Chapter 1

## Introduction

### 1.1 Overview

Computer modeling of blood flows in arteries is an important and very challenging problem. Such simulations can be used to study the human vascular system in a variety of applications, including predicting the development of artery diseases and their treatment [13, 54, 69, 71, 74]. In the early 1990s, computational techniques with finite element methods were applied to simulate blood flows in arteries by using a rigid wall assumption [62, 75]. The results are quite accurate for small arteries, but for large arteries, simulations with such an assumption fail to correctly predict the pressure wave propagation generated by the deformation of the arteries. To improve the accuracy and the understanding of sophisticated hemodynamics in human arteries, fluid-structure interaction (FSI) problems have received more and more attention in recent years. In particular, researchers are increasingly interested in computational techniques for the coupled problem in full three dimensions, that is, all components of the simulation are three-dimensional because these models provide results that can be used to quantify phenomena that are difficult to describe using simplified deformable wall models [26]. Despite the recent advancement of supercomputing technologies, such simulations are still a formidable task because of the difficulties in patient-specific geometry and parameters acquisition, the mathematical formulation and understanding of the coupled system, and the development of efficient parallel solution algorithms that are suitable for high performance computers with a large number of processors.

The contribution of this thesis is the development of a parallel efficient algorithm for the

full three-dimensional FSI problem, with emphasis on strong, monolithic coupling between fluid and structure. Our main concerns in the FSI problem are the accuracy of the physics models, the coupling formulation between the fluid and the structure, the reliability of the discretization, and the parallel performance of the overall solution algorithm. In this introduction, we provide a survey of some recent developments in solving the FSI problem related to blood flow simulations, discuss several algorithms for fluid-structure coupling, and provide an overview of our monolithic coupling approach and parallel solution algorithm.

## 1.2 Modeling fluid-structure interaction problem

Fluid-structure interaction, that is the interaction of a flexible structure with an internal or surrounding fluid flow, is among the most important and the most challenging multi-physics problems. Such interaction gives rise to a rich variety of physical phenomena with applications in many fields of science and engineering. Hence, FSI problems are receiving more and more attention in recent years, and their importance is growing rapidly. For example, FSI problems have been studied in the areas of stability and response of aircraft wings [23, 40], the response of bridges and tall buildings to winds [63, 67], the vibration of turbine and compressor blades [8, 56], and our target application here, the simulation of blood flows in human arteries. To understand these phenomena involving multiple physics we need to model both the structure and the fluid. But FSI problems in general are often too complex to solve analytically, so they have to be analyzed by means of experiments or numerical simulation. Thanks to advances in computational hardware and computational modeling methods and software, more experiments will move from the lab to computer simulations.

Numerical simulations of a viscous incompressible fluid through compliant arteries has many applications such as understanding wave propagation in the artery wall, prediction of areas of turbulence, and acquisition of the artery wall shear stress, all of which are important in the formation and development of the artery diseases. Therefore accurate modeling of the FSI problems can help enabling more reliable prediction of when and where artery disease will occur, and leads to more

timely treatment.

### 1.2.1 Physics of blood flow and artery wall

Blood is a viscous fluid, but containing a significant proportion of non-fluid particles, for example, red blood cells, white blood cells, and platelets. In the human circulation system, blood flow transports oxygen and nutrients to the whole body through the systemic system, and exchange oxygen and carbon dioxide with the lungs through the pulmonary system [61]. Since the viscosity depends on the shear, the behavior of blood is found to be shear-thinning and non-homogeneous. In large arteries, where the shear rate is high, the viscosity is almost constant. Therefore, it is reasonable to consider blood as an incompressible Newtonian fluid in large arteries [64, 76]. While in small vessels, where the shear-thinning property is more apparent, non-Newtonian models [33] are often needed in the simulation. In our work, the focus is on large arteries, blood is therefore considered as an incompressible homogeneous Newtonian fluid and modeled by using the incompressible Navier-Stokes equations.

The artery walls are a complex tissue consisting of different types of constituents, and exhibit viscoelastic characteristics [76]. In practice, most simulations use some simplified viscoelastic models for the artery walls [20, 25]. In recent researches, significant progress has been made in modeling the complex constitutive relation of the artery walls. The nonlinear, anisotropic behavior has been considered in the literatures [7, 39, 52, 95]. Because of our focus, we model the artery walls as a linear elastic structure at this point; moving to a more accurate nonlinear model is a possible direction in the future.

### 1.2.2 Simulating fluid in moving domain

In the simulation of blood flows in compliant arteries, the elastic arterial wall deforms in response to the blood pressure pulse, which in turn implies a moving fluid domain. To keep track of the fluid-structure coupling on the moving interface, different approaches can be used.

In our method, we use an arbitrary Lagrangian-Eulerian (ALE) framework [42, 58] to describe

the fluid equations in the moving domain. Within the ALE framework, the displacement of the fluid domain is introduced as a third field of solution variables in the coupled system. The choice of the governing equations for this field is not based directly on the physics of the fluid-structure problem. The motivation is to provide smooth, well-conditioned elements as the fluid domain deforms at a relatively small computational cost. One possible approach is to make the displacement of the fluid domain satisfy a harmonic extension of the moving fluid-structure interface, as in [5, 6], but it is also possible to model the field as a pseudo-structural system [24, 35, 45]. In the ALE framework, the coupling conditions are guaranteed to be satisfied on the fluid-structure interface. But the new equation for the fluid domain motion and its dependence on the solution introduce further complexities and nonlinearities to the FSI problem.

Space-time formulation [7, 77] is a technique that is closely related to the ALE framework to keep track of the fluid-structure coupling on the interface. In this approach, the computational domain is discretized into time slabs, and a space-time mapping of an arbitrary reference domain is used to describe the mechanics based on the ALE framework.

Another popular approach for simulating fluid in a moving domain is based on the Eulerian framework, namely, the immersed boundary method [30, 84], where the fluid is simulated on a fixed mesh, and the influence of the structure is enforced as a forcing term on the fluid equations. Because of this, the computation of the fluid domain motion is avoided, which makes the immersed boundary methods very competitive because of its simplicity of implementation. But, compared to other boundary-conforming schemes, e.g. the ALE framework, the immersed boundary methods might have decreased order of accuracy because the fluid-structure coupling condition cannot be imposed exactly on the interface [80].

An alternative to the aforementioned approaches is the coupled momentum method [26, 96], in which a membrane model is used to describe the dynamics of the artery wall under the thin wall assumption. The coupled momentum method uses a fixed mesh in the simulation, where the structure momentum contribution is embedded into the fluid equations. These simplifications reduce the computation effort greatly, showing good results in many physiological situations and

demonstrating excellent parallel scalability with a large number of processors. But this method is applicable only for the case of small wall deformation.

### 1.2.3 Coupling formulations for fluid-structure interaction

One of the main challenging issues in solving FSI problems is the coupling algorithm between the fluid and structure subsystems. Two widely used formulations are *iterative* and *monolithic*. In iterative approaches, the fluid and the structure equations are solved one after the other, similar to the nonlinear Gauss-Seidel iteration with two large blocks, updating each other's boundary condition on the interface, until some desired convergence tolerance is reached. In contrast, monolithic approaches couple all subproblems in a single large system. In other words, the fluid, the structure, and the moving domain equations are solved simultaneously as an integrated system, where the coupling conditions are enforced strongly as part of the system.

For many FSI problems, iterative approaches have been successfully employed, including our target application to blood flows in arteries [20, 27]. However, recent studies show difficulties with this kind of algorithms in a number of cases. One example is the so-called added-mass effect, where the convergence of the approach becomes difficult to achieve when similar densities of blood and artery wall are considered [12]. Monolithic approaches [5, 6, 17, 36, 38] tend to have more robust convergence. Many convergence problems encountered in the iterative approaches are eliminated when the fluid and structure are coupled monolithically. Moreover, iterative coupling might also become unstable for large time steps, and in some cases reduce the order of accuracy of the time-stepping algorithm [51], while stability is guaranteed with the monolithic coupling.

### 1.2.4 Boundary conditions

In the blood flow simulation, the size and complexity of the circulation precludes a computational representation for the entire vascular system in human body. Numerical models must invariably be truncated into the upstream domain (domain of interest) and the downstream domain, and boundary conditions must be applied at these cutoff boundaries. Therefore, the choice

of outflow boundary conditions can have a significant influence on the velocity and pressure fields in the simulation. To obtain realistic conditions on these outflow boundaries is not an easy task for a number of reasons. First, because the flow distribution and pressure field in the computational domain are not known, it is very difficult to get the time-varying velocity profile and pressure on each outlet simultaneously from the clinical measurements. Second, since the downstream domain includes a vast quantities of smaller arteries, arterioles, capillaries, venules and veins returning blood to the heart, it might results in inaccurate prediction if we ignore the effects from the absent parts of the circulation system when prescribing the outflow boundary conditions for the modeled domain. It has been reported that, when zero-pressure or zero-traction conditions is utilized on the outlets, the blood pressure might not be computed accurately, and in the case of branching arteries with outlets of different sizes, the flow distribution is computed inconsistently with the results of clinical experiments [26, 82].

Recently, in order to better model the interaction between the computational domain and the downstream vessels, new outflow boundary conditions have been developed. The idea is to use a reduced dimensional model to represent the downstream vessels and provide boundary conditions for the higher dimensional upstream model, where high-resolution information is needed. That is, by taking advantage of known structural patterns in the downstream arterial tree, the relationship between flow and pressure at the outlet boundaries can be enforced by coupling the three-dimensional computational domain with the zero-dimensional, lumped-parameters, or one-dimensional downstream models [27, 28, 59, 60, 70, 79, 81, 82]. Thus, the interaction between the computational domain and the downstream models can be imposed through the outflow boundary conditions, which is crucial in obtaining realistic velocity and pressure fields for the modeled domain.

Among all the choices for outflow boundary conditions, the resistance boundary condition is considered in our application as suggested in [7, 26, 82]. The general concept of resistance is to define a constant relationship between mean pressure and flow rate on the outlets, where we assume the pressure  $P$  is a constant over the upstream outlets. The relation  $P = QR$  is implicitly

prescribed on the outflow boundaries, 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.

Other than the dependence on the accuracy of the 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, the linear Jacobian system is solved by GMRES with a 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. We discuss later in the thesis the impact of the decomposition of the integral condition on the convergence and the scalability of the Schwarz preconditioned Jacobian solver.

### 1.3 Solution algorithms

Depending on the question of how to couple the different physics in the fluid-structure interaction, there are many choices with regard to methods and solvers. In iterative approaches, the coupled FSI system is solved via the partitioned procedure. The coupled system is first partitioned into a fluid and structure subsystem. Then the subproblems are iteratively coupled by a particular choice of transmission conditions on the interface [2, 20, 25, 53]. For many problems, these partitioned schemes work well and are particularly efficient when the well-validated subproblem solvers are available. Solving the monolithically coupled problems is a rather challenging task. The integrated formulation in the approach necessitates a fully coupled solver, precluding the use of existing fluid and structure solvers. We provide below a brief survey of some recent developments in solving fully coupled FSI problems related to blood flow simulations, and finally say a few words

about our fully implicit, monolithic nonlinear solver for the coupled system.

In [77] and references therein, Tezduyar et al. developed a space-time formulation with a stabilized finite element method for fully coupled FSI problems. In their algorithm, the stabilized formulation is based on the streamline-upwind/Petrov-Galerkin (SUPG) and pressure-stabilizing/Petrov-Galerkin (PSPG) methods. The proposed methodology was successfully applied to model blood flows in arteries with aneurysm. In [7], Bazilevs et al. developed a monolithic isogeometric formulation for blood flow simulations in a patient-specific model of an abdominal aortic aneurysm using the space-time methods. The so-called isogeometric analysis based on non-uniform rational B-splines (NURBS) was used to discretize the fully coupled FSI problem in space. Their work is especially suited to smooth curved boundaries and demonstrates very impressive results. But the focus of the aforementioned publications was accuracy, with little attention to computational efficiency and the parallel scalability of the solution algorithms.

Alternatively, least-squares methods were introduced for solving the FSI problems in [14, 15, 37, 38, 47]. In this approach, the entire fluid-structure problem is first reduced to a first-order system of PDEs, and then solved by the minimization of a functional. The resulting system is symmetric positive-definite and that is amenable to solution by multigrid methods. But forming the least-squares functional with these nice characteristics is a difficult task, especially in finding the right boundary conditions for the new variables.

In a more recent publication [17], Crosetto et al. developed a Newton-Krylov algorithm together with a class of preconditioners for solving three-dimensional FSI systems, by taking advantage of the block-structure of a carefully chosen approximate Jacobian system. Their algorithm shows parallel scalability with hundreds of processors, which is quite good, but to obtain high resolution solutions on fine meshes, it is necessary to develop coupled algorithms that are scalable on machines with a much larger number of processors.

In this thesis, we focus on developing a class of parallel Newton-Krylov method with overlapping restricted additive Schwarz preconditioners for solving the fully coupled FSI problems, with emphasis on robustness and parallel scalability of the algorithms. The Newton-Krylov-Schwarz

method has been successfully used in solving various problems and shown good parallel scalability, for example, [34, 44] for fluid dynamics problems, [93] for the atmospheric flow problems, and [5, 6] for two-dimensional FSI problems. But, as far as we know, little work has been done for solving the fully implicit, fully coupled, three-dimensional fluid-structure problems.

In our new method, a stabilized finite element method is introduced and validated for the spatial discretization of the coupled problem and a fully implicit backward difference scheme is used for the temporal discretization. The resulting discretized system is (1) highly nonlinear because of the convective term of the Navier-Stokes equations and the dependency of the solution on the displacement of the moving fluid mesh, and (2) highly unbalanced because of the different characteristics of the fluid subsystem and the structure subsystem. To handle these nonlinearities, in [3, 17], some linearization techniques based on a fixed point algorithm were studied, in which the nonlinear dependence on the moving mesh and/or the convective term is linearized by an extrapolation from the solution of the previous time step. These semi-implicit treatments work well in most cases. In our implementation, we treat all terms in the system implicitly, which leads to a much more stable scheme. We use an inexact Newton method to solve the large nonlinear algebraic system, within which a Krylov subspace method is used to solve the Jacobian systems. Since the parallel scalability of the solution algorithm is mostly determined by how the Jacobian systems are solved, the development of an effective and efficient preconditioner is crucial in the fully implicit solver. In this thesis, we develop a class of monolithic overlapping one-level or two-level additive Schwarz preconditioner to speed up the convergence of a Krylov subspace method.

#### 1.4 Outline of the thesis

The rest of this thesis is organized as follows. In Chapter 2, we describe the formulation of the FSI problem, and the discretization of the coupled problem, both in space and time. In Chapter 3, we present the Newton-Krylov method with one-level and two-level Schwarz preconditioner for solving the fully coupled nonlinear system. Then in Chapter 4 and Chapter 5, we discuss the effectiveness of the algorithm by showing some numerical results using different geometries and

problem sizes, and report the parallel performance of the algorithm. Finally, we provide several concluding remarks and future work in Chapter 6.

## Chapter 2

### Mathematical models and discretizations

When blood flows in an artery, the elastic wall deforms in response to the blood pressure pulse and other possibility external forces, which in turn changes the shape of the fluid domain. Assuming that blood is an incompressible homogeneous Newtonian fluid, we model the fluid-structure interaction by using a coupled system of the linear elasticity equation for the artery and the incompressible Navier-Stokes equations for the blood flows. To address the moving fluid domain, an additional field and the corresponding governing equations are introduced for modeling the domain deformation. In all, the monolithic FSI model is described by three components: the elastic wall structure, the fluid, and the motion of the fluid domain. In this chapter, we present and discuss the models we use for the three fields in the fully coupled problem. Then we derive the weak form of the problem and present the finite element discretization in space and the fully implicit scheme in time.

#### 2.1 Governing equations—strong form

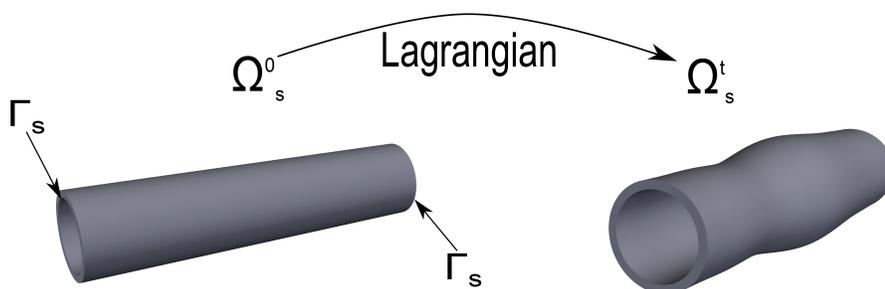
To model the dynamics of the artery wall and the blood, the constitutive relations of structure wall and fluid flows are in general described in the Lagrangian and Eulerian framework, respectively. But in the context of fluid-structure interaction, in order to maintain a Lagrangian description for the structure wall, we consider to use the arbitrary Lagrangian-Eulerian formulation to represent the fluid flows on a moving domain. That is, we first perform a Lagrangian step in order to specify the movement of the fluid domain. Then, we modify the Eulerian description of the fluid equations

Initial configuration

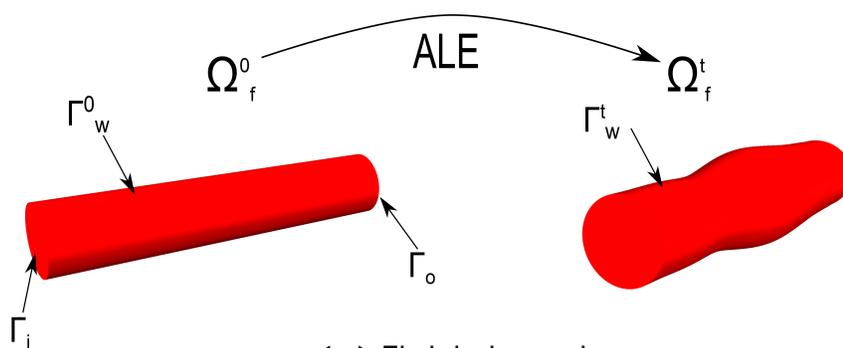
Current configuration



(a) Fluid-structure domain



(b) Structure domain



(c) Fluid domain

Figure 2.1:  $\Omega_s^0$  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,  $\Gamma_i$  and  $\Gamma_o$ , respectively, are fixed and the boundaries  $\Gamma_s$  of the structure domain are also fixed at the inlets and outlets.  $\Gamma_w^t$  represents the fluid-structure interface in current configuration.

with respect to a time-dependent control volume, so that the physical boundary conditions can be specified on the moving fluid-structure interface.

Let  $\Omega^t = \Omega_f^t \cup \Omega_s^t$  be the combined fluid subdomain  $\Omega_f^t \subset R^3$ , and structure subdomain  $\Omega_s^t \subset R^3$  at time  $t$ . The initial configuration of the domain is defined as  $\Omega^0 = \Omega_f^0 \cup \Omega_s^0$  when  $t = 0$ .  $\Gamma_w^t = \partial\Omega_f^t \cap \partial\Omega_s^t$  represents the interface between the fluid and structure subdomains, and  $\Gamma_w^0$  is the corresponding interface in the initial configuration. See Figure 2.1 for a schematic and some notations.

### 2.1.1 Structure model

We model the structure problem with a linear elasticity equation, using the Lagrangian frame of reference. The displacement  $\mathbf{x}_s$  of the structure at the Lagrangian configuration is assumed to satisfy

$$\rho_s \frac{\partial^2 \mathbf{x}_s}{\partial t^2} + \alpha \frac{\partial \mathbf{x}_s}{\partial t} - \nabla \cdot \sigma_s = \mathbf{f}_s \quad \text{in } \Omega_s^0, \quad (2.1)$$

where  $\rho_s$  is the density of the structure,  $\alpha$  is a mass-proportional damping parameter, and the Cauchy stress tensor  $\sigma_s$  is given by

$$\sigma_s = \lambda_s (\nabla \cdot \mathbf{x}_s) I + \mu_s (\nabla \mathbf{x}_s + \nabla \mathbf{x}_s^T).$$

The Lamé parameters  $\lambda_s$  and  $\mu_s$  are properties of the physical material under consideration, which are related to the Young's modulus  $E$  and the Poisson ratio  $\nu_s$  by

$$\lambda_s = \frac{\nu_s E}{(1 + \nu_s)(1 - 2\nu_s)}, \quad \mu_s = \frac{E}{2(1 + \nu_s)}.$$

In [5, 6], two-dimensional blood flows in compliant arteries were successfully simulated without the stabilization of the elasticity equation; i.e.,  $\alpha = 0$ . In other words, the impact of surrounding tissues is not considered in 2D, and the instability problem does not show up. However, according to numerical experiments conducted by us and others, in three-dimensional simulations, the blood flow and the elasticity waves inside the artery may become unstable sometimes, without a carefully

chosen stability constant  $\alpha$ . Following [72, 77], a mass-proportional damping coefficient  $\alpha$  is considered in our formulation to represent the damping effect of the surrounding tissue on the artery. Other choices concerning the surrounding tissue effects can also be found in [18], which imposes special boundary conditions on the external artery walls.

As for the boundary conditions, we fix the displacement of the structure on the inlet and outlets. Such constraints may not be a realistic representation of the physical situation. But since it is not practical to model the structural characteristics of an entire human body, these conditions are popular choices for the cut-off boundaries, and have been used in [5, 7, 26]. Also, we use zero-Neumann conditions on the external boundaries of the structure.

### 2.1.2 Moving domain description

To model the fluid in the moving domain  $\Omega_f^t$ , we use an arbitrary Lagrangian-Eulerian formulation. Consider an ALE mapping  $A_t$  defined from the reference configuration  $\Omega_f^0$  to the moving domain  $\Omega_f^t$ :

$$A_t : \Omega_f^0 \rightarrow \Omega_f^t, \quad \mathbf{x}(Y, t) = A_t(\mathbf{Y}), \quad \forall \mathbf{Y} \in \Omega_f^0,$$

where  $\mathbf{Y}$  is referred to as the ALE coordinates and  $\mathbf{x}$  as the Eulerian coordinates. A generic conservation law defined on a moving domain  $\Omega_f^t$  given by

$$\frac{\partial u}{\partial t} + \nabla_{\mathbf{x}} \cdot F(u) = f$$

can be written in the ALE form as

$$\left. \frac{\partial u}{\partial t} \right|_{\mathbf{Y}} - \omega_g \cdot \nabla_{\mathbf{x}} u + \nabla_{\mathbf{x}} \cdot F(u) = f, \quad (2.2)$$

where  $\omega_g = \partial A_t / \partial t$  is the velocity of the moving domain and  $\mathbf{Y}$  indicates that the time derivative is taken with respect to the ALE coordinates.

To specify the ALE formulation for the FSI problem, we maintain the Lagrangian description for the structure wall, and use it to provide a boundary condition for the movement of the fluid domain. That is, we assume that the displacement of the fluid domain  $\mathbf{x}_f$  at the reference

configuration  $\Omega_f^0$  satisfies a harmonic extension of the moving fluid-structure interface,

$$\Delta \mathbf{x}_f = 0 \quad \text{in } \Omega_f^0. \quad (2.3)$$

Next, we define an ALE mapping  $A_t$  from  $\Omega_f^0$  to  $\Omega_f^t$ :

$$A_t : \Omega_f^0 \rightarrow \Omega_f^t, \quad A_t(\mathbf{Y}) = \mathbf{Y} + \mathbf{x}_f(\mathbf{Y}), \quad \forall \mathbf{Y} \in \Omega_f^0, \quad (2.4)$$

where  $\mathbf{Y}$  is referred to as the ALE coordinates. In general, the choice of the model for the moving fluid domain (2.3) is not unique and not based directly on the physics of the FSI problem. Other choices are available [35, 45]. Based on our experiments, this simple scheme performs well, maintaining good conditioning of elements even under relatively large deformation.

The boundary condition for this field are fixed zero Dirichlet conditions at the inlet and outlets of the fluid domain. That is

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

In addition, the displacement at the fluid-structure interface  $\Gamma_w$  are required to follow the movement of the structure, so that fluid domain deformation matches structure movement.

### 2.1.3 Fluid model

The incompressible Navier-Stokes equations defined on the moving domain  $\Omega_f^t$  are rewritten in the ALE form by the generic conservation law (2.2) and the definition of the mapping  $A_t$  (2.4) as

$$\begin{aligned} \rho_f \frac{\partial \mathbf{u}_f}{\partial t} \Big|_{\mathbf{Y}} + \rho_f [(\mathbf{u}_f - \omega_g) \cdot \nabla] \mathbf{u}_f - \nabla \cdot \sigma_f &= 0 \quad \text{in } \Omega_f^t, \\ \nabla \cdot \mathbf{u}_f &= 0 \quad \text{in } \Omega_f^t, \end{aligned}$$

where  $\rho_f$  is the fluid density,  $\mathbf{u}_f$  is the fluid velocity,  $\sigma_f = -p_f I + \mu_f (\nabla \mathbf{u}_f + \nabla \mathbf{u}_f^T)$  is the Cauchy stress tensor,  $\omega_g = \partial \mathbf{x}_f / \partial t$  is the velocity of the moving domain, and  $\mathbf{Y}$  indicates that the time derivative is taken with respect to the ALE coordinates.

We need to specify the boundary conditions for the fluid equations. On the inlet boundaries  $\Gamma_i$ , a given velocity profile is prescribed as a Dirichlet boundary condition. On the outlet boundaries  $\Gamma_o$ , the resistance boundary condition is considered. The idea is to use a reduced dimensional model to represent the downstream vessels and provide boundary condition for the domain of interest [26, 82, 81]. The resistance is to define a constant relationship between the pressure and the flow on the outlet boundaries,

$$p_f = RQ = R \int_{\Gamma_o} \mathbf{u}_f \cdot \mathbf{n} \, ds \quad \text{on } \Gamma_o, \quad (2.5)$$

where  $\mathbf{n}$  is the unit outward normal and  $R$  is measured resistance. In some cases, we also consider the zero traction boundary condition at the outlets,

$$\sigma_f \cdot \mathbf{n} = (-p_f + \mu_f(\nabla \mathbf{u}_f + \nabla \mathbf{u}_f^T)) \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_o. \quad (2.6)$$

#### 2.1.4 Coupling conditions

In addition to the equations above, three coupling conditions are needed at the fluid-structure interface. First of all, we require the continuity of the velocities on the interface,

$$\mathbf{u}_f = \frac{\partial \mathbf{x}_s}{\partial t}. \quad (2.7)$$

Secondly, we require the continuity of the traction forces on the interface,

$$\sigma_s \cdot \mathbf{n}_s = -\sigma_f \cdot \mathbf{n}_f, \quad (2.8)$$

where  $\mathbf{n}_s$ ,  $\mathbf{n}_f$  are unit normal vectors for the structure and fluid domains. At last, we require that the motion of the fluid domain follows the structure displacement, so that the structure can maintain a Lagrangian description,

$$\mathbf{x}_f = \mathbf{x}_s. \quad (2.9)$$

To summarize, the strong form of our fully coupled FSI problem is given by

$$\begin{aligned}
\rho_s \frac{\partial^2 \mathbf{x}_s}{\partial t^2} + \alpha \frac{\partial \mathbf{x}_s}{\partial t} - \nabla \cdot \sigma_s &= \mathbf{f}_s && \text{in } \Omega_s, \\
\mathbf{x}_s &= 0 && \text{on } \Gamma_s, \\
\rho_f \frac{\partial \mathbf{u}_f}{\partial t} \Big|_Y + \rho_f [(\mathbf{u}_f - \omega_g) \cdot \nabla] \mathbf{u}_f - \nabla \cdot \sigma_f &= 0 && \text{in } \Omega_f(t), \\
\nabla \cdot \mathbf{u}_f &= 0 && \text{in } \Omega_f(t), \\
\mathbf{u}_f &= g && \text{on } \Gamma_i, \\
p_f = RQ = R \int_{\Gamma_o} \mathbf{u}_f \cdot \mathbf{n} \, ds &&& \text{on } \Gamma_o, \\
\Delta \mathbf{x}_f &= 0 && \text{in } \Omega_0, \\
\mathbf{x}_f &= 0 && \text{on } \Gamma_i \cup \Gamma_o, \\
\sigma_s \cdot \mathbf{n}_s &= -\sigma_f \cdot \mathbf{n}_f && \text{on } \Gamma_w, \\
\mathbf{u}_f &= \frac{\partial \mathbf{x}_s}{\partial t} && \text{on } \Gamma_w, \\
\mathbf{x}_f &= \mathbf{x}_s && \text{on } \Gamma_w.
\end{aligned}$$

## 2.2 Governing equations—weak form

To derive the weak form of the structure equation, we define the variational space of the structure problem as

$$X = \{ \mathbf{x}_s \in [H^1(\Omega_s^0)]^3 : \mathbf{x}_s = 0 \text{ on } \Gamma_s \}.$$

Following the standard procedure, we multiply (2.1) by a test function  $\phi_s$  and integrate over the domain to get,

$$\rho_s \int_{\Omega_s^0} \frac{\partial^2 \mathbf{x}_s}{\partial t^2} \cdot \phi_s \, d\Omega + \alpha \int_{\Omega_s^0} \frac{\partial \mathbf{x}_s}{\partial t} \cdot \phi_s \, d\Omega - \int_{\Omega_s^0} (\nabla \cdot \sigma_s) \cdot \phi_s \, d\Omega = \int_{\Omega_s^0} \mathbf{f}_s \cdot \phi_s \, d\Omega$$

Integrating by parts on the third term of the above equation, we have

$$\int_{\Omega_s^0} (\nabla \cdot \sigma_s) \cdot \phi_s \, d\Omega = \int_{\Gamma_w^0} \phi_s \cdot (\sigma_s \cdot \mathbf{n}_s) \, ds - \int_{\Omega_s^0} a(\phi_s) \cdot Da(\mathbf{x}_s) \, d\Omega \quad (2.10)$$

where

$$a(\mathbf{u}) = \begin{pmatrix} \frac{\partial u_1}{\partial x} \\ \frac{\partial u_2}{\partial y} \\ \frac{\partial u_3}{\partial z} \\ \frac{\partial u_2}{\partial z} + \frac{\partial u_3}{\partial y} \\ \frac{\partial u_1}{\partial z} + \frac{\partial u_3}{\partial x} \\ \frac{\partial u_1}{\partial y} + \frac{\partial u_2}{\partial x} \end{pmatrix}, \quad D = \begin{pmatrix} \lambda_s + 2\mu_s & \lambda_s & \lambda_s & 0 & 0 & 0 \\ \lambda_s & \lambda_s + 2\mu_s & \lambda_s & 0 & 0 & 0 \\ \lambda_s & \lambda_s & \lambda_s + 2\mu_s & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu_s & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu_s & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu_s \end{pmatrix}.$$

Here  $\mathbf{u} = (u_1, u_2, u_3)^T$  and  $D$  is a matrix of constant that describe the material properties of the structure, which are related to the Lamé parameters  $\lambda_s$  and  $\mu_s$ . Hence the Cauchy stress tensor  $\sigma_s$  can be given by

$$\sigma_s = Da(\mathbf{x}_s).$$

It is also important to emphasize that the coupling condition (2.8) is implicitly enforced as part of (2.10) by the relation

$$\int_{\Gamma_w^0} \phi_s \cdot (\sigma_s \cdot \mathbf{n}_s) \, ds + \int_{\Gamma_w^t} \phi_f \cdot (\sigma_f \cdot \mathbf{n}_f) \, ds = 0, \quad (2.11)$$

where the test function  $\phi_f$  is defined below.

As a result, the corresponding weak form of the structure problem is stated as follows: Find  $\mathbf{x}_s \in X$  such that  $\forall \phi_s \in X$ ,

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

with

$$\begin{aligned} B_s(\mathbf{x}_s, \phi_s; \sigma_f) &= \rho_s \int_{\Omega_s^0} \frac{\partial^2 \mathbf{x}_s}{\partial t^2} \cdot \phi_s \, d\Omega + \alpha \int_{\Omega_s^0} \frac{\partial \mathbf{x}_s}{\partial t} \cdot \phi_s \, d\Omega + \int_{\Omega_s^0} a(\phi_s) \cdot Da(\mathbf{x}_s) \, d\Omega \\ &+ \int_{\Gamma_w^t} \phi_f \cdot (\sigma_f \cdot \mathbf{n}_f) \, ds - \int_{\Omega_s^0} \mathbf{f}_s \cdot \phi_s \, d\Omega. \end{aligned} \quad (2.12)$$

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 (2.7).

We define the trial and weighting function spaces as

$$\begin{aligned} V &= \{ \mathbf{u}_f \in [H^1(\Omega_f^t)]^3 : \mathbf{u}_f = g \text{ on } \Gamma_i, \mathbf{u}_f = \partial \mathbf{x}_s / \partial t \text{ on } \Gamma_t^w \}, \\ V_0 &= \{ \mathbf{u}_f \in [H^1(\Omega_f^t)]^3 : \mathbf{u}_f = 0 \text{ on } \Gamma_i \cup \Gamma_t^w \}, \\ P &= L^2(\Omega_f^t). \end{aligned}$$

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

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

with

$$\begin{aligned} B_f(\{\mathbf{u}_f, p_f\}, \{\phi_f, \psi_f\}; \omega_g) &= \rho_f \int_{\Omega_f^t} \frac{\partial \mathbf{u}_f}{\partial t} \Big|_Y \cdot \phi_f \, d\Omega - \int_{\Omega_f^t} p_f (\nabla \cdot \phi_f) \, d\Omega \\ &+ \rho_f \int_{\Omega_f^t} [(\mathbf{u}_f - \omega_g) \cdot \nabla] \mathbf{u}_f \cdot \phi_f \, d\Omega + 2\mu_f \int_{\Omega_f^t} \epsilon(\mathbf{u}_f) : \epsilon(\phi_f) \, d\Omega \\ &+ \int_{\Omega_f^t} (\nabla \cdot \mathbf{u}_f) \psi_f \, d\Omega + \int_{\Gamma_o} (\sigma_f \cdot \mathbf{n}) \cdot \phi_f \, ds. \end{aligned} \quad (2.13)$$

Here  $\epsilon(\mathbf{u}_f) = (\nabla \mathbf{u}_f + \nabla \mathbf{u}_f^T)/2$ . The last term in (2.13), which represents the contribution of the resistance boundary condition on the outlet  $\Gamma_o$ , can be rewritten as

$$\int_{\Gamma_o} (\sigma_f \cdot \mathbf{n}) \cdot \phi_f \, ds = - \int_{\Gamma_o} \left( R \int_{\Gamma_o} \mathbf{u}_f \cdot \mathbf{n} \, ds \right) \phi_f \cdot \mathbf{n} \, ds + 2\mu_f \int_{\Gamma_o} \phi_f \cdot \epsilon(\mathbf{u}_f) \cdot \mathbf{n} \, ds \quad (2.14)$$

thanks to the relation (2.5). If the zero-traction boundary condition is considered, the last term in (2.13) can be omitted. We remark that the fluid problem couples the fluid domain displacement  $\mathbf{x}_f$  through the term  $\omega_g$  and the integration domain, and connects to the structure velocity  $\partial \mathbf{x}_s / \partial t$  by the coupling condition (2.7).

The weak form of the domain movement problem reads as follows: Find  $\mathbf{x}_f \in Z$  such that  $\forall \phi_m \in Z_0$ ,

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

with

$$B_m(\mathbf{x}_f, \phi_m) = \int_{\Omega_f^0} \nabla \mathbf{x}_f : \nabla \phi_m \, d\Omega, \quad (2.15)$$

where the variational spaces are defined as

$$\begin{aligned} Z &= \{\mathbf{x}_f \in [H^1(\Omega_f^0)]^3 : \mathbf{x}_f = \mathbf{x}_s \text{ on } \Gamma_w^0, \mathbf{x}_f = 0 \text{ on } \Gamma_i \cup \Gamma_o\}, \\ Z_0 &= \{\mathbf{x}_f \in [H^1(\Omega_f^0)]^3 : \mathbf{x}_f = 0 \text{ on } \Gamma_i \cup \Gamma_o \cup \Gamma_w^0\}. \end{aligned}$$

### 2.3 Spatial discretization

We discretize the weak problem in space with a conforming finite element method. All the spatial discretization is done with tetrahedral finite elements. The mesh is generated in a way so that the interface between the fluid and structure is aligned with the mesh; see Figure 2.2. In other words, the interface does not cut through any elements. But such a restriction is not followed later when we partition the mesh to define the domain decomposition solver. Moreover, with this conforming discretization at the fluid-structure interface, no special interpolation scheme is necessary to move information between fluid and structure. The finite element subspaces we considered consists of quasi-uniform unstructured  $P_1 - P_1$  stabilized elements for the fluid,  $P_1$  elements for the structure, and  $P_1$  elements for the fluid domain motion. In the remainder of this section, we 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.

#### 2.3.1 Stabilized finite element method

The finite element approximation of weak forms for the structure (2.12) and fluid domain motion (2.15) is straight forward. But since the discretization of the fluid problem in the ALE framework involves the discrete form of fluid domain motion, the finite element approximation of weak form (2.13) needs more work. Suppose  $\mathcal{T}_f^{h,0}$  is the given unstructured tetrahedral mesh of the reference fluid configuration  $\Omega_f^0$ . The mesh  $\mathcal{T}_f^{h,t}$  of the current configuration  $\Omega_f^t$  is defined through the discrete ALE mapping  $A_t$ , where  $\mathcal{T}_f^{h,t} = A_t(\mathcal{T}_f^{h,0})$ . In our implementation, the discrete ALE

mapping comes from the equation of fluid domain movement, which is defined as an harmonic extension of the moving fluid-structure interface.

To discretize the incompressible Navier-Stokes equations, the standard Galerkin method is well known to be unstable for two reasons. The first is due to the numerical instabilities that might be encountered when we have high Reynolds number and strong boundary layers. The second is the pressure stability related to the limitation of the choices of the finite element interpolations used to approximate the velocity and pressure fields. The latter issue requires the pair of finite element subspaces  $V_h$  and  $P_h$  to satisfy the LBB inf-sup condition:

$$\inf_{q_h \in P_h} \sup_{u_h \in V_h} \frac{|\int_{\Omega} \nabla \cdot u_h q_h d\Omega|}{\|u_h\|_{H^1} \|q_h\|_{L^2}} \geq \beta,$$

where the constant  $\beta$  is independent of the discretization size  $h$ . For example, the Taylor-Hood finite element  $P_k - P_{k-1}$  is one of the famous inf-sup stable pairs. That is, the velocity is approximated by the polynomial of degree  $k$ , while the pressure is approximated by polynomial of degree  $k - 1$ . On the other hand, in the case of equal-order interpolation (e.g  $P_1 - P_1$  pair), an alternative is to use the stabilized methods to circumvent the limit of LBB inf-sup condition by modifying the variational formulation. Typical stabilization techniques include the streamline-upwind/Petrov-Galerkin (SUPG) [9, 29], Galerkin-least-squares (GLS) [41] and the orthogonal subgrid scales (OSS) [16] formulations to name a few.

Among all the possible choices, we consider the SUPG based stabilization formulation, where additional stabilization terms are used in the formulation with equal-order interpolation of the velocity and the pressure as described in [75, 86]. The semi-discrete stabilized finite element formulation for the fluid problem reads as follows: Find  $\mathbf{u}_f \in V_h$  and  $p_f \in P_h$ , such that  $\forall \phi_f \in V_{h,0}$  and  $\forall \psi_f \in P_h$ ,

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

with

$$\begin{aligned}
& B(\{\mathbf{u}_f, p_f\}, \{\phi_f, \psi_f\}; \omega_g) \\
&= B_f(\{\mathbf{u}_f, p_f\}, \{\phi_f, \psi_f\}; \omega_g) + \sum_{K \in \mathcal{T}_f^{h,t}} (\nabla \cdot \mathbf{u}_f, \tau_c \nabla \cdot \phi_f)_K \\
&+ \sum_{K \in \mathcal{T}_f^{h,t}} \left( \frac{\partial \mathbf{u}_f}{\partial t} \Big|_Y + (\mathbf{u}_f - \omega_g) \cdot \nabla \mathbf{u}_f + \nabla p_f, \tau_m ((\mathbf{u}_f - \omega_g) \cdot \nabla \phi_f + \nabla \psi_f) \right)_K \\
&+ \sum_{K \in \mathcal{T}_f^{h,t}} (\bar{\mathbf{u}}_f \cdot \nabla \mathbf{u}_f, \phi_f)_K + \sum_{K \in \mathcal{T}_f^{h,t}} (\bar{\mathbf{u}}_f \cdot \nabla \mathbf{u}_f, \tau_b \bar{\mathbf{u}}_f \cdot \nabla \phi_f)_K,
\end{aligned}$$

where  $\mathcal{T}_f^{h,t} = \{K\}$  is the given unstructured tetrahedral fluid mesh, and  $\bar{\mathbf{u}}_f$  is the conservation-restoring advective velocity introduced in [75],

$$\bar{\mathbf{u}}_f = -\tau_m \left( \frac{\partial \mathbf{u}_f}{\partial t} \Big|_Y + (\mathbf{u}_f - \omega_g) \cdot \nabla \mathbf{u}_f + \nabla p_f \right).$$

The stabilization parameters  $\tau_m$ ,  $\tau_c$ , and  $\tau_b$  are defined as in [7] and similar stabilization parameters are used in [26, 85] for some problems defined on a fixed mesh, where

$$\begin{aligned}
\tau_m &= \frac{1}{\sqrt{\frac{4}{\Delta t^2} + (\mathbf{u}_f - \omega_g) \cdot G(\mathbf{u}_f - \omega_g) + 36 \left(\frac{\mu_f}{\rho_f}\right)^2 G : G}}, \\
\tau_c &= \frac{1}{8\tau_m \text{tr}(G)}, \\
\tau_b &= \frac{1}{\sqrt{\bar{\mathbf{u}}_f \cdot G \bar{\mathbf{u}}_f}}.
\end{aligned}$$

Here,  $G_{ij} = \sum_{k=1}^3 \frac{\partial \xi_k}{\partial x_i} \frac{\partial \xi_k}{\partial x_j}$  denotes the covariant metric tensor, which may be identified with the element length scale [85], and  $\frac{\partial \xi}{\partial \mathbf{x}}$  represents the inverse Jacobian of the mapping between the reference and the physical domains. The term  $4/\Delta t^2$  in  $\tau_m$  is important only for time dependent problems, and is dropped for steady-state computations.

### 2.3.2 Semi-discrete fully coupled problem

We form the finite dimensional fully coupled FSI problem as follows: Find  $\mathbf{x}_s \in X_h$ ,  $\mathbf{u}_f \in V_h$ ,  $p_f \in P_h$  and  $\mathbf{x}_f \in Z_h$  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  $\forall \phi_m \in Z_{h,0}$ ,

$$B_s(\mathbf{x}_s, \phi_s; \sigma_f) + B(\{\mathbf{u}_f, p_f\}, \{\phi_f, \psi_f\}; \omega_g) + B_m(\mathbf{x}_f, \phi_m) = 0, \quad (2.16)$$

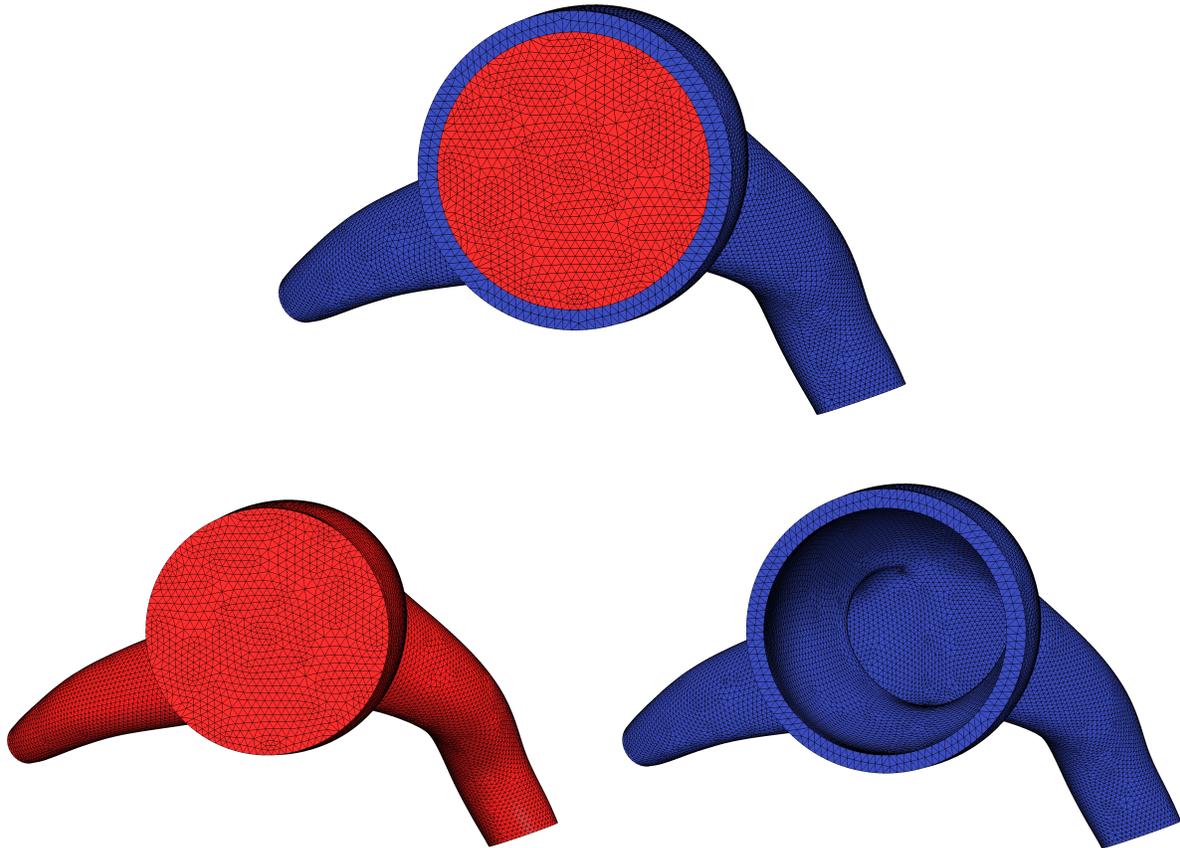


Figure 2.2: Example finite element mesh for the fluid-structure interaction problem (top). The mesh is generated so that the interface between the fluid and structure is aligned with the mesh. The elements of the fluid (bottom left) and the elements of the structure (bottom right) conform on the interface.

where the discrete versions of coupling conditions (2.7) and (2.9) are enforced directly at each time step. That is the certain degrees of freedom are set to be equal to each other at the fluid-structure interface. While the traction coupling condition (2.8) is prescribed in the boundary term of  $B_s$  as described in (2.12).

Let  $\{\varphi_i\}_{i \in I_s}$  be the finite element basis functions for  $X_h$ . We write the finite element approximation  $\mathbf{x}_s$  of the structure displacement as

$$\mathbf{x}_s = \sum_i x_i(t) \varphi_i(x).$$

Using this approximation, the algebraic form of the first term  $B_s$  in (2.16) becomes

$$M_s \frac{\partial^2 x_s}{\partial t^2} + \alpha M_s \frac{\partial x_s}{\partial t} + K_s x_s + F_s = 0 \quad (2.17)$$

Here, we denote the vector of coefficients  $(x_j)$  as  $x_s$ , and the matrices  $M_s$  and  $K_s$  are defined as follows

$$\begin{aligned} (M_s)_{ij} &= \rho_s \int_{\Omega_s^0} \varphi_j \cdot \varphi_i \, d\Omega, \\ (K_s)_{ij} &= \int_{\Omega_s^0} a(\varphi_j) \cdot Da(\varphi_i) \, d\Omega. \end{aligned}$$

The  $F_s$  term in (2.17) represents the body and surface forces,

$$(F_s)_i = \int_{\Gamma_w^t} (\sigma_f \cdot \mathbf{n}_f) \cdot \phi_i \, ds - \int_{\Omega_s^0} \mathbf{f}_s \cdot \phi_i \, d\Omega. \quad (2.18)$$

Note that the  $\sigma_f \cdot \mathbf{n}_f$  in the above equation comes from the weak formulation of the coupling condition (2.11) on the fluid-structure interface, which acts as a force on the structure boundary.

Discretization of such force requires integration of the appropriate fluid terms on the fluid-structure interface. Let  $\phi_i$  and  $\psi_i$  be the finite element basis of the fluid velocity and pressure, respectively.

We have

$$\sigma_f \cdot \mathbf{n}_f = - \left( \sum_j p_j \psi_j \right) \mathbf{n}_f + 2\mu_f \left( \sum_j u_j \epsilon(\phi_j) \right) \cdot \mathbf{n}_f.$$

Then the equation (2.18) can be rewritten as

$$(F_s)_i = - \sum_j p_j \left( \int_{\Gamma_w^t} (\psi_j \mathbf{n}_f) \cdot \phi_i \, ds \right) + 2\mu_f \sum_j u_j \left( \int_{\Gamma_w^t} (\epsilon(\phi_j) \cdot \mathbf{n}_f) \cdot \phi_i \, ds \right) - \int_{\Omega_s^0} \mathbf{f}_s \cdot \phi_i \, d\Omega.$$

On the other hand, let  $\{\xi_i\}_{i \in I_f}$  be the finite element basis for  $Z_h$ . We write the finite element approximations  $\mathbf{x}_f$  of the fluid domain displacement as

$$\mathbf{x}_f = \sum_j x_j(t) \xi_j(x).$$

Using this approximation, the algebraic form of the third term  $B_m$  in (2.16) becomes

$$K_m x_f = 0, \tag{2.19}$$

where  $x_f = (x_j)$  and the matrix  $K_m$  is introduced as

$$(K_m)_{ij} = \int_{\Omega_f^0} \xi_j \cdot \xi_i \, d\Omega.$$

As for the discretization for the fluid problem, we denote  $\{\phi_i\}_{i \in I_f}$  and  $\{\psi_i\}_{i \in I_f}$  as the finite element basis functions for  $V_h$  and  $P_h$ , respectively. The finite element approximations  $\mathbf{u}_f$  and  $\mathbf{p}_f$  are defined as

$$\mathbf{u}_f = \sum_i u_i(t) \phi_i(x, t), \quad \mathbf{p}_f = \sum_i p_i(t) \psi_i(x, t),$$

where  $\phi$  and  $\psi$  depend on  $t$  because of the moving ALE mesh. Moreover, we denote the discrete vector of coefficient  $u_f = (u_i)$ , and  $p_f = (p_i)$ .

First, to consider the time derivative in the ALE framework of  $\mathbf{u}_f$ , we have

$$\frac{\partial \mathbf{u}_f}{\partial t} \Big|_Y = \sum_i \left( u_j \frac{\partial \phi_i}{\partial t} \Big|_Y + \frac{du_i}{dt} \Big|_Y \phi_i \right) = \sum_i \frac{du_i}{dt} \phi_i,$$

by the definition of the ALE derivative

$$\frac{\partial \phi_i}{\partial t} \Big|_Y = \frac{\partial \hat{\phi}_i}{\partial t} = 0,$$

where  $\hat{\phi}_i$  is defined as a basis function on the time-invariant reference configuration  $\Omega_f^0$ .

Then, the algebraic form of the second term  $B$  in (2.16) becomes

$$M_f(\mathbf{u}_f; \omega_g) \frac{du_f}{dt} + N_f(\mathbf{u}_f; \bar{\mathbf{u}}_f; \omega_g) u_f - Q_f^T(\mathbf{u}_f; \omega_g) p_f + F_f = 0 \quad (2.20)$$

$$S_f(\mathbf{u}_f; \omega_g) \frac{du_f}{dt} + Q_f(\mathbf{u}_f; \omega_g) u_f + R_f(\mathbf{u}_f; \omega_g) p_f = 0 \quad (2.21)$$

where

$$\begin{aligned} (M_f(\mathbf{u}_f; \omega_g))_{ij} &= \rho_f \int_{\Omega_f^t} \phi_j \cdot \phi_i \, d\Omega + \sum_{K \in \mathcal{T}_f^{h,t}} \int_K \phi_j \cdot (\tau_m(\mathbf{u}_f - \omega_g) \cdot \nabla \phi_i) \, d\Omega, \\ (N_f(\mathbf{u}_f; \bar{\mathbf{u}}_f; \omega_g))_{ij} &= \rho_f \int_{\Omega_f^t} [(\mathbf{u}_f - \omega_g) \cdot \nabla] \phi_j \cdot \phi_i \, d\Omega + 2\mu_f \int_{\Omega_f^t} \epsilon(\phi_j) : \epsilon(\phi_i) \, d\Omega \\ &\quad + \sum_{K \in \mathcal{T}_f^{h,t}} \int_K ((\mathbf{u}_f - \omega_g) \cdot \nabla \phi_j) \cdot (\tau_m(\mathbf{u}_f - \omega_g) \cdot \nabla \phi_i) \, d\Omega \\ &\quad + \sum_{K \in \mathcal{T}_f^{h,t}} \int_K (\nabla \cdot \phi_j) \cdot (\tau_c \nabla \cdot \phi_i) \, d\Omega + \sum_{K \in \mathcal{T}_f^{h,t}} \int_K (\bar{\mathbf{u}}_f \cdot \nabla \phi_j) \cdot \phi_i \, d\Omega \\ &\quad + \sum_{K \in \mathcal{T}_f^{h,t}} \int_K (\bar{\mathbf{u}}_f \cdot \nabla \phi_j) \cdot (\tau_b \bar{\mathbf{u}}_f \cdot \nabla \phi_i) \, d\Omega, \\ (Q_f(\mathbf{u}_f; \omega_g))_{ij} &= \int_{\Omega_f^t} (\nabla \cdot \phi_j) \psi_i \, d\Omega + \sum_{K \in \mathcal{T}_f^{h,t}} \int_K \tau_m ((\mathbf{u}_f - \omega_g) \cdot \nabla \phi_j) \cdot \nabla \psi_i \, d\Omega, \\ (S_f(\mathbf{u}_f; \omega_g))_{ij} &= \sum_{K \in \mathcal{T}_f^{h,t}} \int_K \phi_j \cdot (\tau_m \nabla \psi_i) \, d\Omega, \\ (R_f(\mathbf{u}_f; \omega_g))_{ij} &= \sum_{K \in \mathcal{T}_f^{h,t}} \int_K \nabla \psi_j \cdot (\tau_m \nabla \psi_i) \, d\Omega, \\ (F_f)_i &= \int_{\Gamma_o} (\sigma_f \cdot \mathbf{n}) \cdot \phi_i \, ds. \end{aligned}$$

Here, the last term  $F_f$  in the above derivation represents the contribution of surface integral on the outlet boundaries. When the resistance boundary condition is considered on  $\Gamma_o$ , we have

$$(F_f)_i = - \sum_j u_j \left( \int_{\Gamma_o} \phi_j \cdot \mathbf{n} \, ds \right) \left( R \int_{\Gamma_o} \phi_i \cdot \mathbf{n} \, ds \right) + 2\mu_f \sum_j u_j \int_{\Gamma_o} \phi_i \cdot \epsilon(\phi_j) \cdot \mathbf{n} \, ds,$$

by the relation (2.14). In this case, the pressure and velocity on the boundary become unknown solution variables, which are enforced as an implicitly coupled boundary condition.

Since we model the fluid in the ALE framework, the integration domain in (2.20) and (2.21) depends implicitly on the fluid mesh displacement  $x_f$ . So that the operators  $M_f$ ,  $N_f$ ,  $Q_f$ ,  $S_f$  and  $R_f$  all depend on  $x_f$  implicitly through the time-dependent integration domain. In addition,  $M_f$ ,  $N_f$ ,  $Q_f$  also depend on  $x_f$  explicitly through the mesh velocity term  $\omega_g$  that appears in the formulation. Furthermore, the operators  $M_f$ ,  $N_f$ ,  $Q_f$ ,  $S_f$  and  $R_f$  have dependence on the fluid velocity  $u_f$ . Such nonlinear dependence first come through the convective term directly, and then through the stabilization parameters  $\tau_m$ ,  $\tau_c$ ,  $\tau_b$  implicitly. Lastly,  $N_f$  also depends on the conservation-restoring advective velocity  $\bar{\mathbf{u}}_f$ , which will introduce further nonlinear dependence to  $u_f$ ,  $p_f$  and  $x_f$ .

To conclude this section, we remark that the finite element formulations for each of the three subproblems have been validated. One of the goals of the thesis is to validate, through numerical experiments, the correctness of the coupled formulation, including all three components, as well as the coupling conditions. The second goal of the thesis is to develop an algorithm for solving the discretized system.

## 2.4 Fully implicit time discretization

Our time discretization for the fully coupled FSI problem is fully implicit. In the FSI system, the governing equations of the fluid problem is first-order in time, while the elasticity equation of the structure problem is second-order in time. By introducing the structure velocity  $\dot{x}_s$  as an additional unknown variable, we first reduce the second-order equation (2.17) to a first-order system of equations as in [31]

$$\begin{aligned} \frac{\partial x_s}{\partial t} - \dot{x}_s &= 0, \\ M_s \frac{\partial \dot{x}_s}{\partial t} + \alpha M_s \dot{x}_s + K_s x_s + F_s &= 0. \end{aligned}$$

Then, denote the vector of unknowns that includes both the structure displacement and velocity as

$$y_s = \begin{pmatrix} x_s \\ \dot{x}_s \end{pmatrix},$$

we write the first-order system of equation in the matrix form as

$$\begin{pmatrix} I & 0 \\ 0 & M_s \end{pmatrix} \frac{\partial y_s}{\partial t} + \begin{pmatrix} 0 & -I \\ K_s & \alpha M_s \end{pmatrix} y_s + \begin{pmatrix} 0 \\ F_s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Therefore, we have the semi-discrete form of the entire monolithically coupled system (2.16) as

$$M \frac{dy}{dt} + Ky + F = 0 \quad (2.22)$$

with

$$y = \begin{pmatrix} u_f \\ p_f \\ x_f \\ x_s \\ \dot{x}_s \end{pmatrix}, \quad M = \begin{pmatrix} M_f(\mathbf{u}_f; \omega_g) & 0 & 0 & 0 & 0 \\ S_f(\mathbf{u}_f; \omega_g) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & M_s \end{pmatrix}, \quad F = \begin{pmatrix} F_f \\ 0 \\ 0 \\ 0 \\ F_s \end{pmatrix}.$$

$$K = \begin{pmatrix} N_f(\mathbf{u}_f; \bar{\mathbf{u}}_f; \omega_g) & -Q_f^T(\mathbf{u}_f; \omega_g) & 0 & 0 & 0 \\ Q_f(\mathbf{u}_f; \omega_g) & R_f(\mathbf{u}_f; \omega_g) & 0 & 0 & 0 \\ 0 & 0 & K_m & 0 & 0 \\ 0 & 0 & 0 & -I & 0 \\ 0 & 0 & 0 & 0 & K_s \quad \alpha M_s \end{pmatrix}.$$

By using the same time-stepping scheme for both the fluid and the structure, the semi-discrete system (2.22) is further discretized in time with a second-order backward differentiation formula (BDF2). That is, for a given semi-discrete system

$$\frac{dy}{dt} = \mathcal{L}(y),$$

the BDF2 scheme

$$y^n - \frac{4}{3}y^{n-1} + \frac{1}{3}y^{n-2} = \frac{2\Delta t}{3}\mathcal{L}(y^n)$$

is employed for the time integration, where  $y^n$  represents the value of  $y$  at the  $n^{\text{th}}$  time step with a fixed time step size  $\Delta t$ . This temporal discretization scheme is fully implicit. At each time step, we obtain the solution  $y^n = (u_f, p_f, x_f, x_s, \dot{x}_s)$  at the  $n^{\text{th}}$  time step from the previous two time steps by solving a sparse, nonlinear algebraic system

$$\mathcal{F}_n(y^n) = 0, \quad (2.23)$$

with

$$\mathcal{F}_n(y^n) = M \left( y^n - \frac{4}{3}y^{n-1} + \frac{1}{3}y^{n-2} \right) + \frac{2\Delta t}{3}Ky^n + \frac{2\Delta t}{3}F. \quad (2.24)$$

Here,  $y^n$  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  $n^{\text{th}}$  time step. For simplicity, we ignore the script  $n$  in (2.23) for the rest of the thesis. Note that the discretization of the fluid problem in the ALE framework includes the mesh velocity  $\omega_g$ , which also needs to be discretized in time. We calculate the mesh velocity by using the first order divided differences of the mesh displacement at each time step, that is

$$\omega_g = \frac{x_f^n - x_f^{n-1}}{\Delta t}.$$

Though written in the algebraic form (2.23), the operators  $M$  and  $K$  in (2.24) are in fact depend on the current solution, therefore, are nonlinear. Since our time discretization scheme is fully implicit, all the terms of the equations are treated implicitly. As a result,  $\mathcal{F}$  is highly nonlinear, where the nonlinearities come from the convective term of the Navier-Stokes equations, the stabilization terms, and the dependency on the displacement of the moving fluid mesh as shown in Section 2.3.2. In some recent publications, certain linearization techniques based on fixed point algorithms are sometimes considered in the time discretization [3, 17], in which the nonlinear dependence on the moving mesh or the convective term are linearized by extrapolating

the information from the previous time step. Such semi-implicit time discretizations are good in most situations, but may not be stable when the time step size is large. The fully implicit approach is more stable and more robust, but the corresponding nonlinear systems are quite difficult to solve because of the different characteristics its components have. In the fluid part of  $\mathcal{F}$ , there are 4 unknowns per mesh point; in the structure part, there are 6 unknowns per mesh point; and in the moving mesh part, there are 3 unknowns per mesh point. The equations for the fluid is time dependent, nonlinear parabolic-like; the equations for the structure are time dependent, linear hyperbolic-like; and the equations for the moving mesh are elliptic type. The equations for the interface conditions are enforced implicitly as part of the system. The stiffness of the system is different in the fluid part and the structure part of the computational domain, depending on the viscosity coefficient of the flow and the wall.

## Chapter 3

### A monolithic nonlinear solver for the coupled system of equations

#### 3.1 Overview of Newton-Krylov-Schwarz methods

In this chapter, we introduce a class of overlapping domain decomposition methods for solving the coupled multi-physics system (2.23). The method is well studied for each individual component of the problem, namely, the incompressible Navier-Stokes equation [43, 44], the linear elasticity equation [49], and the elliptic moving mesh equation [11]. In recent papers [5, 6], it was extended to the coupled system in two-dimensional space; here, we further extend it to a full three-dimensional problem. To design an algorithm for (2.23) that is highly scalable in terms of the total compute time, many important factors need to be taken into consideration. The basic components of the algorithm are not new, but to arrive at the best combination, we consider not only the properties of the nonlinear system, the properties of the domain decomposition methods, but also the software and hardware of our computational environment. In this work, one of the main ideas is the tight, monolithic coupling between fluid and structure. We use a Newton-Krylov-Schwarz algorithm to solve the fully coupled FSI system. In the Newton-Krylov-Schwarz approach, the nonlinear system (2.23) is solved via the inexact Newton method [21, 22]. That is, at each Newton step the new solution  $x^{(k+1)}$  is obtained from the current solution  $x^{(k)}$  by

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

where the step length  $\theta^{(k)}$  is determined by a cubic line search technique [19]. The Newton correction  $s^{(k)}$  is approximated by solving a right-preconditioned Jacobian system with a Krylov subspace

method, GMRES [66]

$$J_k M_k^{-1} M_k s^{(k)} = -\mathcal{F}(x^{(k)}), \quad (3.1)$$

where  $J_k$  is the Jacobian matrix evaluated at  $x^{(k)}$  and  $M_k^{-1}$  is a one-level or two-level additive Schwarz preconditioner. Within the inexact Newton method, the accuracy of the solution to the Jacobian system is controlled by the forcing term  $\eta_k$  to ensure

$$\|J_k M_k^{-1} M_k s^{(k)} + \mathcal{F}(x^{(k)})\| \leq \eta_k \|\mathcal{F}(x^{(k)})\|.$$

Note that, a flexible version of GMRES (fGMRES) [65] has to be used if the preconditioner changes during the GMRES iterations. This happens when the coarse problems in a two-level preconditioner are solved iteratively.

### 3.2 Evaluation of Jacobian

In Newton-Krylov methods, Newton methods can be implemented with or without the explicit form of the Jacobian matrix. Since the Jacobian is only needed in the form of matrix-vector multiplications, this can be done approximately in a matrix-free approach [50] using

$$J(x)y \approx \frac{\mathcal{F}(x + \epsilon y) - \mathcal{F}(x)}{\epsilon},$$

where  $\epsilon$  is small. In our method, we do not take the matrix-free approach because the explicit form of the Jacobian is required in constructing the preconditioner. The evaluation of the Jacobian of the fully coupled system is non-trivial, especially for three-dimensional problems, as most researchers choose to approximate the Jacobian by ignoring certain terms. The difficulty lies in the evaluation of the cross derivatives; e.g., the derivatives of the fully coupled system with respect to the mesh movement. One solution is to use a finite difference approximation to calculate the cross derivatives [36], but such approximation is required at each Newton iteration and may drastically increase the overall compute time. Another solution is to use a computationally inexpensive approximation of the Jacobian [32], but this may deteriorate the overall convergence. In our implementation, we

compute the Jacobian analytically, including all those cross derivatives. There are 66 derivatives at some of the grid points, so the task of hand-calculating these derivatives is time-consuming. However, this is a worthwhile exercise since it saves many Newton iterations, and provides a better preconditioner for the Jacobian systems. We remark that the robustness of Newton method is often not guaranteed when the Jacobian is approximately computed.

### 3.3 One-level restricted additive Schwarz preconditioner

A critically important component of the overall solver is the preconditioner, without which the iterative Jacobian solver (3.1) would not converge well and, as a result, the outer inexact Newton may not converge well either. To define the one-level Schwarz preconditioner, we first partition the finite element mesh  $\mathcal{T}_h$ , constructed for the initial configuration, into non-overlapping subdomains  $\Omega_\ell^h$ ,  $\ell = 1, \dots, N$ , where the number of subdomain  $N$  is always the same as the number of processors  $np$ . The partition is element-based, and does not distinguish if an element is a fluid or a structure element, etc. The degrees of freedom defined at a mesh point are taken into account to ensure load balancing; i.e., each subdomain has more or less the same number of unknowns. Then each subdomain  $\Omega_\ell^h$  is extended to an overlapping subdomain  $\Omega_\ell^{h,\delta}$ ; see Figure 3.1. Here,  $\delta$  is an integer indicating the level of overlap. We assume that the boundary  $\partial\Omega_\ell^h$  does not cut any elements of  $\mathcal{T}_h$  and the union of  $\Omega_\ell^h$  covers the entire mesh  $\mathcal{T}_h$ . In addition, the extension of the non-overlapping subdomains also respects element boundaries, so that each  $\Omega_\ell^h$  and  $\Omega_\ell^{h,\delta}$  consists of an integral number of elements. We remark that the decomposition of the mesh is completely independent of which physical variables are defined for a given mesh point. A subdomain may contain both fluid and structure elements. Another criterion of the partition is to minimize the edge cuts for the purpose of saving communication cost.

On each subdomain  $\Omega_\ell^{h,\delta}$ , we decompose the boundary  $\partial\Omega_\ell^{h,\delta}$  into the physical boundary  $\Gamma_\ell^P$  and the artificial boundary  $\Gamma_\ell^A$ , where  $\Gamma_\ell^A = \partial\Omega_\ell^{h,\delta} \setminus \Gamma_\ell^P$ . On  $\Gamma_\ell^P$ , we impose the corresponding physical boundary conditions according to the model, while on  $\Gamma_\ell^A$  we assume the homogeneous Dirichlet boundary conditions for all variables. The well-posedness of the boundary condition is

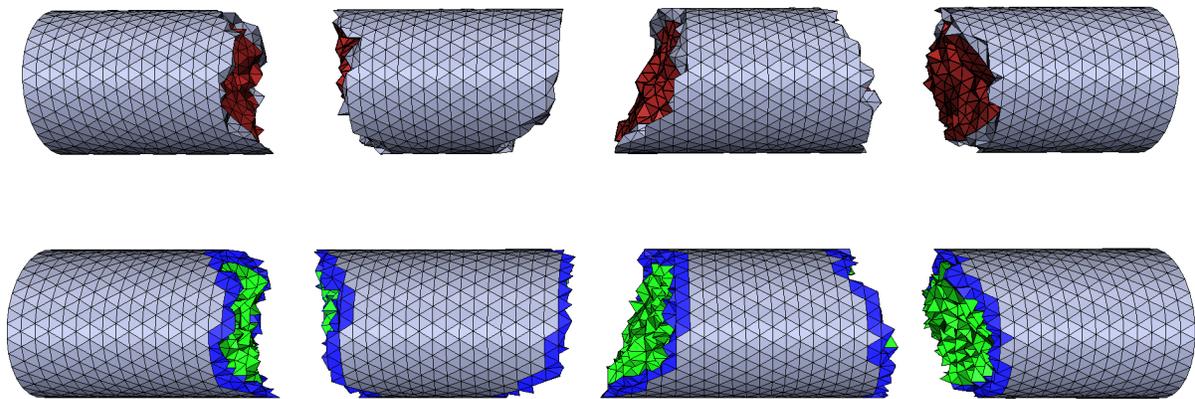


Figure 3.1: Sample partition of the domain into 4 subdomains by using ParMETIS. The fluid elements and structure elements are marked with different colors. The top figure shows the partition into non-overlapping subdomains, and the bottom represents a corresponding partition into overlapping subdomains with  $\delta = 2$ . The shaded elements in blue and green represent the corresponding fluid elements and structure elements extended from the non-overlapping subdomains.

not known, but numerical experiments suggest that the setup is acceptable, at least for the class of problems under consideration. We define the solution space  $S^h$  as

$$S^h = \{(u_f, p_f, x_f, x_s, \dot{x}_s) : u_f \in V_h, p_f \in P_h, x_f \in Z_h, x_s \in X_h, \dot{x}_s \in X_h\},$$

and the subdomain solution space  $S_\ell^{h,\delta}$  as

$$\begin{aligned} S_\ell^{h,\delta} = \{ & (u_f, p_f, x_f, x_s, \dot{x}_s) : u_f \in V_h \cap [H^1(\Omega_\ell^{h,\delta})]^3, u_f = 0 \text{ on } \Gamma_\ell^A; \\ & p_f \in P_h \cap L^2(\Omega_\ell^{h,\delta}), p_f = 0 \text{ on } \Gamma_\ell^A; \\ & x_f \in Z_h \cap [H^1(\Omega_\ell^{h,\delta})]^3, x_f = 0 \text{ on } \Gamma_\ell^A; \\ & x_s \in X_h \cap [H^1(\Omega_\ell^{h,\delta})]^3, x_s = 0 \text{ on } \Gamma_\ell^A; \\ & \dot{x}_s \in X_h \cap [H^1(\Omega_\ell^{h,\delta})]^3, \dot{x}_s = 0 \text{ on } \Gamma_\ell^A\}. \end{aligned}$$

Let  $R_\ell : S^h \rightarrow S_\ell^{h,\delta}$  be a restriction operator, which returns all degrees of freedom associated with the subspace  $S_\ell^{h,\delta}$ . We define a subdomain Jacobian by  $B_\ell = R_\ell J_k R_\ell^T$ , which is a restriction of the Jacobian matrix to the subdomain  $J_k$ . The one-level restricted additive Schwarz preconditioner is given by

$$M_{one}^{-1} = \sum_{\ell=1}^N (R_\ell^0)^T B_\ell^{-1} R_\ell, \quad (3.2)$$

where  $R_\ell^0$  is the restriction to the degrees of freedom in the non-overlapping subdomain  $\Omega_\ell^h$ . In the restricted additive Schwarz preconditioner [11], the overlapping regions between the overlapping subdomains are used to provide information to the subdomain solve, but the results of computation in the overlapping regions are not considered in the prolongation procedure in order to reduce the communication cost when implemented on parallel computers.

### 3.3.1 Subdomain solver

On each subdomain, sparse LU factorization based direct methods can be used to solve the subdomain Jacobian system. But LU factorization can be computationally expensive if the subdomain problem is large, which often happens when the number of processors is relatively small.

To improve the efficiency of such factorization, we replace it with some versions of incomplete LU factorization. There are two types of ILU factorizations. The popular pointwise ILU works well for matrices arising from scalar partial differential equations, but sometimes fails to work for coupled multi-physics problems. We choose to use some point-block versions of ILU as the subdomain solve, where we group all physical components associated with a mesh point as a block. By using the point-block version, we can considerably improve the robustness of the subdomain preconditioner, and at the same time improve the cache performance of the computation. In implementation, the inverse of the small point-block matrix on the diagonal of the large matrix is computed exactly before the ILU factorization is carried out.

### 3.4 Some theory of Schwarz methods

The convergence theory for one-level and two-level Schwarz methods is very well developed for elliptic problems; see [68, 78]. In particular, for elliptic systems with sufficient shape regularity, the condition number  $\kappa$  of the preconditioned operator satisfies

$$\kappa \leq \frac{C(1 + H/\delta)}{H^2}$$

for the one-level preconditioner and

$$\kappa \leq C(1 + H/\delta)$$

for the two-level preconditioner, where  $H$  is the subdomain diameter,  $\delta$  is the overlap size, and the constant  $C$  is independent of  $H$ ,  $\delta$  and the discretization size  $h$ . The factor  $1/H^2$ , which is proportional to the number of subdomains, in the one-level method indicates that the preconditioning effect becomes weaker and the number of iteration increases as we increase the number of processors. The two-level method is often preferred in order to remove the dependency of the number of iterations on the number of processors.

In the case of time-dependent parabolic problems, the condition number in the one-level preconditioners remains bounded as long as the time step is not too large. The condition number

of the preconditioned operator acts like

$$\kappa \leq C \left( 1 + \frac{\Delta t}{H\delta} \right)$$

for the parabolic one-level ASM with overlap [10]. In addition, for the time-dependent scalar hyperbolic time-dependent case, it is known that the condition number can be made independent of  $h$ ,  $H$ , and  $\Delta t$  if the overlap is sufficient [87].

Since our problem is a complicated system of mixed type, it is not clear how much of the preceding theory applies. In practice, we do see that the number of linear iterations increases as we increase the number of subdomains for the one-level method [90, 91]. Although this increase is not as great as the elliptic theory predicts, the scaling of our problem is still not so nice as the number of subdomains increases. As an attempt to fix this, and also to reduce the number of linear iterations more generally, we have implemented the two-level methods, which we will say more in the next section.

### 3.5 Two-level Schwarz preconditioner

The basic idea of the two-level Schwarz method is to include a coarse mesh in the preconditioner, in order to facilitate exchange of information between different subdomains. For example, if the global problem is partitioned into  $N$  subdomains, then in the one-level method it might take  $N$  iterations for information to get from a subdomain to the one that is located farthest away. Including a coarse space in the preconditioner allows the transfer of this information in a single iteration.

#### 3.5.1 Interpolation and restriction

To formally define the hybrid two-level Schwarz preconditioner, we first need to specify the choice of the coarse mesh and the partition based on that. In our setting, the fine mesh does not have to be a refinement of the coarse mesh. And the coarse mesh inherits the partition of the fine mesh; see Figure 3.2 for an example. The information exchange between the coarse and fine meshes is through the coarse-to-fine interpolation matrix  $I_H^h$ . The construction of the interpolation matrix

$I_H^h$  is based on finite elements. For each degree of freedom  $v_i$  on the fine mesh, it is extrapolated from the coarse mesh by

$$v_i \approx (I_H^h v^H)_i = \sum_j v_j^H \phi_j^H(x_i), \quad (3.3)$$

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  $\phi_j^H$  is the finite element basis function of the  $j$ th degree of freedom on the coarse mesh. With an unstructured mesh, it may occasionally happen that a fine mesh point is not contained within any coarse element. In this situation, interpolation is done on the closest coarse element.

In our simulation, the fluid mesh deforms through the fluid-structure interaction, so the interpolation matrix should be updated at each time step. But, since we do not have explicit access to the basis function  $\phi_j$  on an unstructured mesh, the calculation of the basis function is costly. For simplicity, we construct the interpolation matrix only based on the reference configuration.

### 3.5.2 Parallel implementation

We employ a hybrid two-level preconditioner, defined by combining the coarse-level preconditioner  $M_H^{-1}$  and the fine-level preconditioner  $M_h^{-1}$  multiplicatively:

$$\begin{aligned} M_{two}^{-1} = & M_h^{-1} + M_H^{-1} + M_h^{-1} - M_h^{-1} J_h M_h^{-1} \\ & - M_h^{-1} J_h M_H^{-1} - M_H^{-1} J_h M_h^{-1} + M_h^{-1} J_h M_H^{-1} J_h M_h^{-1}, \end{aligned} \quad (3.4)$$

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 (3.2). On the coarse-level, the preconditioning  $M_H^{-1} = I_H^h J_H^{-1} (I_H^h)^T$  is constructed by solving a smaller linear system associated with the coarse Jacobian matrix  $J_H$ .

More precisely, the application of the two-level hybrid Schwarz preconditioner  $y = M_{two}^{-1} x$  is obtained in the following three steps: we first apply the fine-level preconditioner,

$$y = \left( \sum_{\ell=1}^N (R_\ell^0)^T B_\ell^{-1} R_\ell \right) x; \quad (3.5)$$

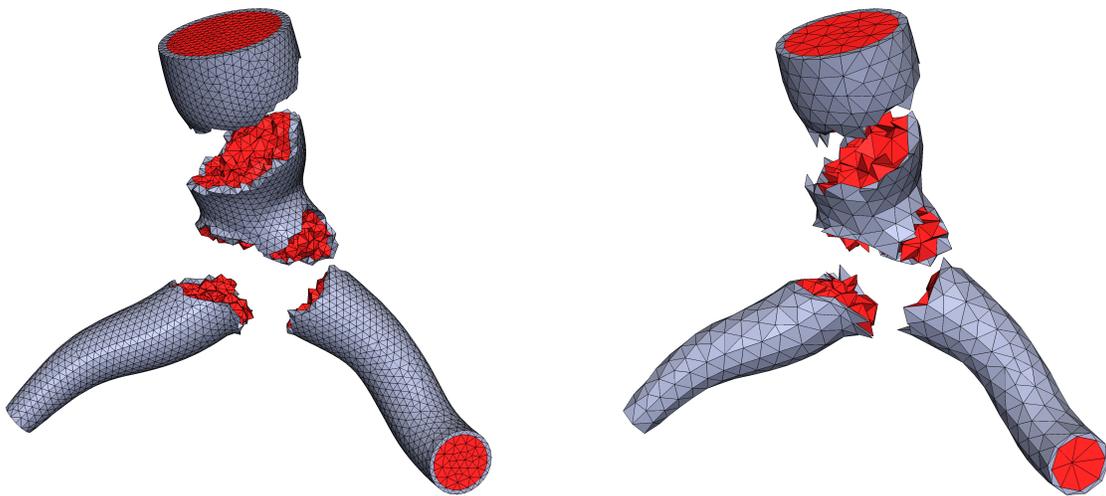


Figure 3.2: Example partitions of the fine mesh and the coarse mesh of a pulmonary artery by using ParMETIS. The fluid element and structure elements are marked with different colors. The left figure shows the partition of the fine mesh, and the right represents a corresponding partition of the coarse mesh. The elements of a coarse subdomain are assigned to the corresponding processor that contains the fine mesh elements from the same subdomain.

next, we update the solution by adding the coarse-level correction,

$$y = y + I_H^h J_H^{-1} (I_H^h)^T (x - J_h y); \quad (3.6)$$

and finally, we apply the fine-level preconditioner one more time,

$$y = y + \left( \sum_{\ell=1}^N (R_\ell^0)^T B_\ell^{-1} R_\ell \right) (x - J_h y). \quad (3.7)$$

Note that in the cascade multigrid approach, the operation (3.5) can be omitted before the coarse mesh preconditioning (3.6). We define the cascade two-level Schwarz precondition as

$$M_{cas}^{-1} = M_H^{-1} + M_h^{-1} - M_h^{-1} J_h M_H^{-1}. \quad (3.8)$$

The cascade type of two-level preconditioners has been successfully employed in many application, including shallow water equations [92] and 2D FSI problems [6]. But in the case of 3D FSI problems, our experiments suggest that there is a benefit to include the second swipe of the one-level preconditioner to the preconditioner as in (3.4).

### 3.5.3 Coarse-level solver

The coarse-level preconditioning by  $M_H^{-1} = I_H^h J_H^{-1} (I_H^h)^T$  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 the same way we solve the fine problem. That is we solve the linear problem iteratively by using GMRES with a one-level restricted additive Schwarz preconditioner, with the same number of subdomains as in the fine mesh. Using the full parallel collective in solving the coarse problem efficient use of the parallel computers. But one potential disadvantage is the large number of subdomains on the coarse space, which could lead to the same ill-conditioning problem that the one-level methods have. In practice, the coarse problem is usually easy enough to solve and the overlap can be made sufficiently large to overcome this difficulty. Another possible solution is to reduce the number of subdomains used in coarse level preconditioner and let the rest of processors sit idle during the coarse solve, but this is not most

efficient in the use of the parallel machine. A more sophisticated solution would be to add more levels to the preconditioner as in [55, 92].

## Chapter 4

### 2D FSI simulations with resistance boundary condition

In this chapter, we study the application of our solver to the two-dimensional FSI model with resistive outflow boundary conditions. Although not completely physically realistic for blood flow simulation, this simplified 2D problem maintains most of the mathematical aspects of the fully coupled FSI problem with the integral type resistive outflow boundary conditions. Therefore it is a worthwhile test case before moving to full three-dimensional simulation. In the following, we first study the impact of the resistance outflow boundary conditions on the accuracy of the hemodynamic prediction. We then investigate the numerical behavior and parallel performance of the Schwarz preconditioners with this integral type resistive boundary conditions.

Since a two-dimensional problem is considered in this chapter, we rewrite the solid momentum equation (2.1) to include the  $\gamma$  term, which is used to represent a radially symmetric artery in two dimensions [3, 58]:

$$\rho_s \frac{\partial^2 \mathbf{x}_s}{\partial t^2} + \alpha \frac{\partial \mathbf{x}_s}{\partial t} - \nabla \cdot \sigma_s + \gamma \mathbf{x}_s = \mathbf{f}_s \quad \text{in } \Omega_s^0.$$

We use the finite element method to discretize the 2D fully coupled fluid-structure problem in space, with mixed  $Q_2 - Q_1$  elements for the fluid and  $Q_2$  elements for the structure. And we discretize in time with the second order implicit trapezoid rule. Detailed derivations of this 2D fully coupled problem with resistive boundary conditions including the weak formulation and discretizations can be found in [88, 89].

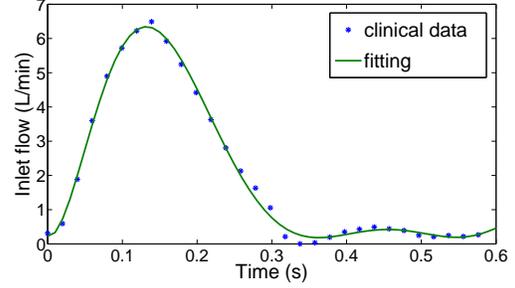
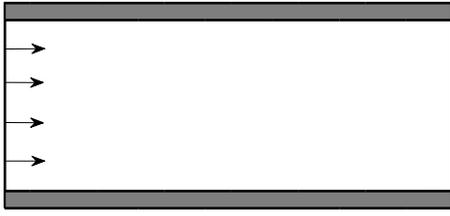


Figure 4.1: The setup of the 2D straight tube problem and the inlet flow rate from clinical data with a polynomial fitting.

## 4.1 Impact of different outflow boundary conditions

### 4.1.1 2D straight tube

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 and 2 cm in diameter, and the artery wall is 0.2 cm in thickness. For the inlet, we prescribe a pulsatile periodic flow wave,<sup>1</sup> with a period  $T$  of 0.6 s, mapped to a uniform velocity profile; see Figure 4.1. For the outlet, both zero-traction and resistance boundary conditions are considered. In the case of the resistance boundary condition, the resistance  $R = 599.85 \text{ dyn} \cdot \text{s}/\text{cm}^5$  is prescribed on the outlet. The elastic artery wall is characterized with density of  $1.2 \text{ g}/\text{cm}^3$ , Young's modulus of  $6.0 \times 10^5 \text{ g}/(\text{cm s}^2)$  and Poisson ratio of 0.48. The blood is modeled as a Newtonian flow, with a density of  $1.0 \text{ g}/\text{cm}^3$  and viscosity of  $0.035 \text{ g}/(\text{cm s})$ . The Reynolds number is approximately 300 based on the mean inlet velocity and the radius of the artery. We initialize the simulation by setting the initial wall velocity to zero and use the solution of the steady state FSI problem as the initial conditions for the unsteady problem.

A mesh with 58369 elements and 1.01 million degrees of freedom is utilized and the solutions are obtained with a time step of 1 ms for a total of 3 cardiac cycles. The simulation proceeds to the next time step when the residual of the nonlinear system is less than  $10^{-6}$ . In Figure 4.2, we

<sup>1</sup> The data is provided by Z. Su, K. Hunter, and R. Shandas of University of Colorado, School of Medicine.

compare the results obtained with two types of outflow boundary conditions, resistance and zero-traction. 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 result in unrealistic amplitude and oscillatory pattern.

Figure 4.3 displays the computed streamlines for the resistance boundary condition at two different phase, the peak systole, and mid-diastole ( $t = 5T/6$ ). We can see the deformation of artery walls in response to the pulse of the the flow. At the peak systole phase, the artery walls dilate so that the flow at the outlet damps. While at the mid-diastole phase, the flow is slower, the artery walls shrink and more flow is leaving than entering. We also observe that at the mid-diastole stage, some vortices are generated near the inlet boundary and the direction of the flow is reversed near the artery wall. Figure 4.4 shows the computed streamlines for the zero traction boundary condition at the same phases. Compared to the resistance case, the flow pattern behaves differently and the wall deformation is smaller. At the mid-diastole phase, reserve flow is observed near the outlet boundary.

#### 4.1.2 2D bifurcating artery

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 Figure 4.5, one of the branches stenoses with a 75% area reduction. We use the same inlet boundary conditions, and the same material properties for the blood and vessel wall as in the previous test. The finite element mesh consists of 65920 elements and 1.15 million of unknowns. The simulations are run for 3 cardiac

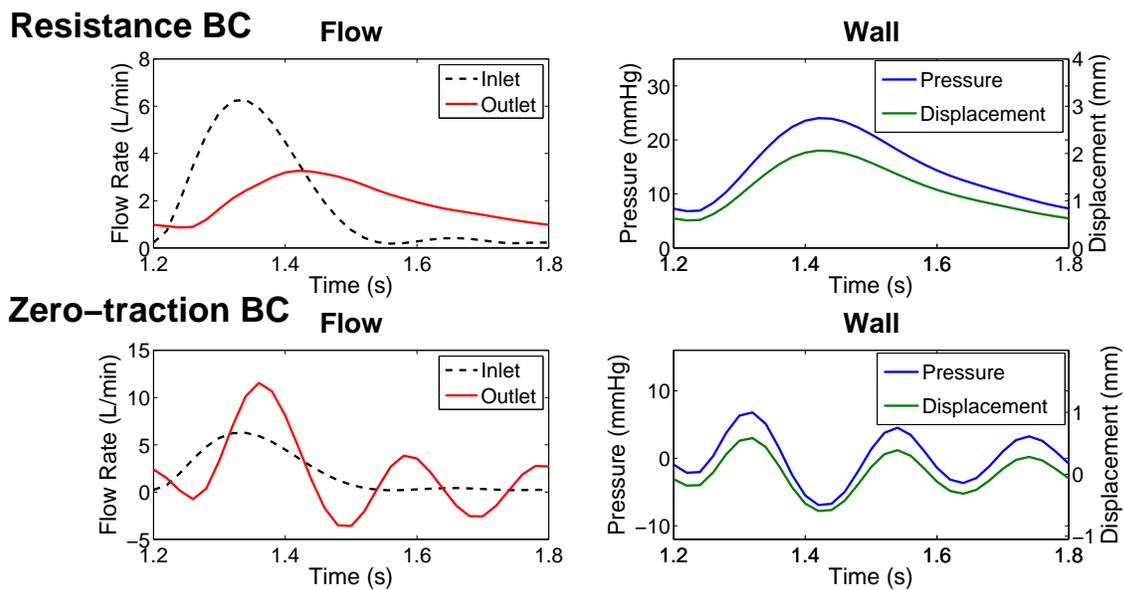


Figure 4.2: Flow waves at inlet and outlet, wall pressure and displacement over one cardiac cycles, obtained using resistance and zero-traction outlet boundary conditions for the 2D straight tube problem. Top left figure represents the inlet and outlet flow rate, and top right figure shows the wall displacement and pressure using the resistance boundary condition, while the bottom figures represent the results using the zero-traction outflow boundary condition.

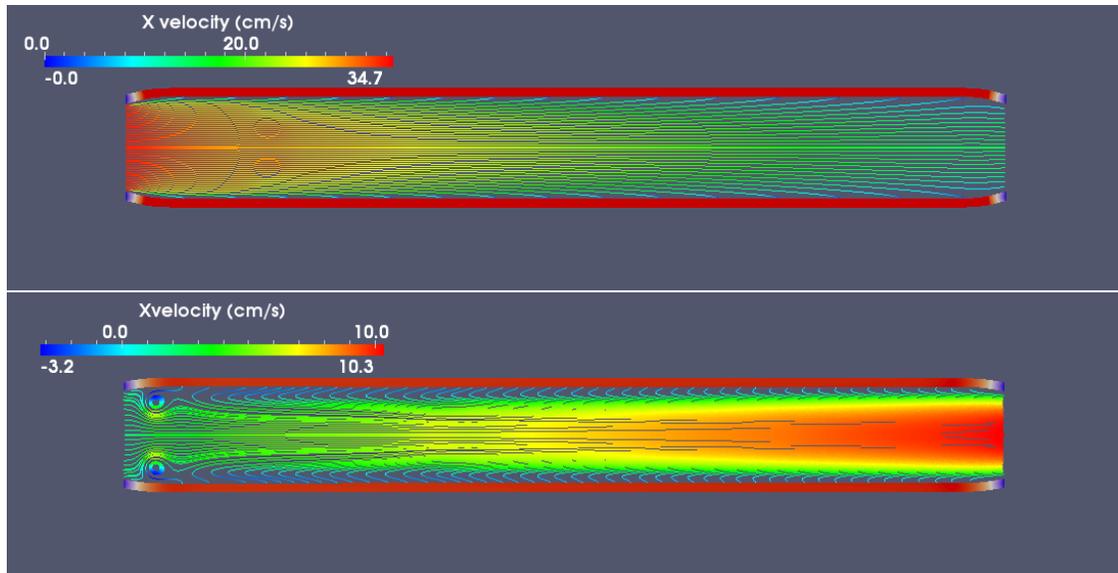


Figure 4.3: Streamlines at the peak systole and mid-diastole phases for the 2D straight tube problem using the resistance outflow boundary condition. The artery walls are colored by the magnitude of the displacement and fluid streamlines are colored by the velocity in the X direction. The top figure represents the peak systole phase, the bottom figure for the mid-diastole phase.

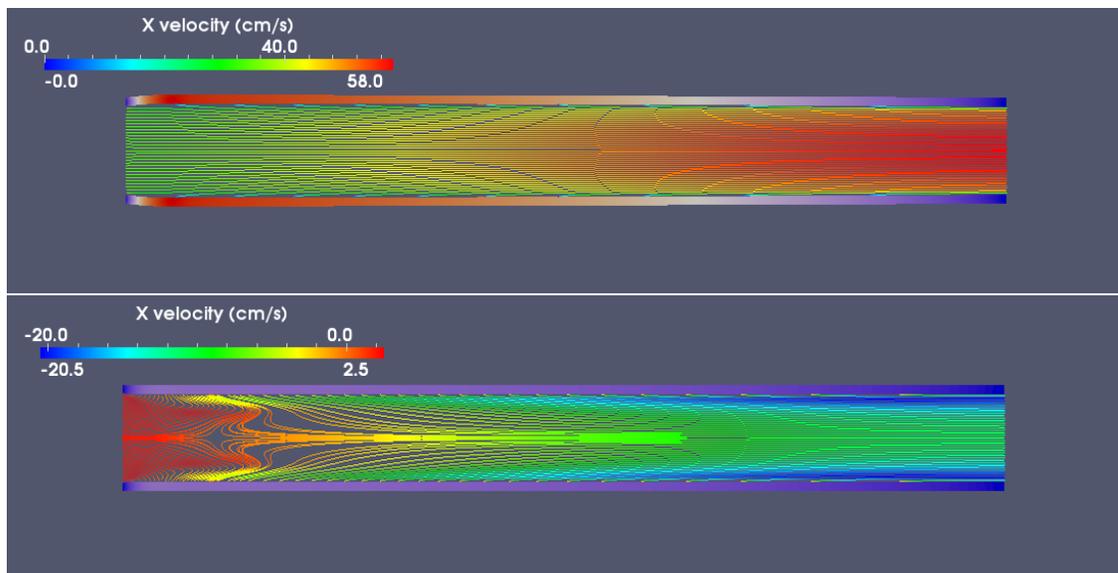


Figure 4.4: Streamlines at the peak systole and mid-diastole phases for the 2D straight tube problem using the zero-traction outflow boundary condition. The artery walls are colored by the magnitude of the displacement and fluid streamlines are colored by the velocity in the X direction. The top figure represents the peak systole phase, the bottom figure for the mid-diastole phase.

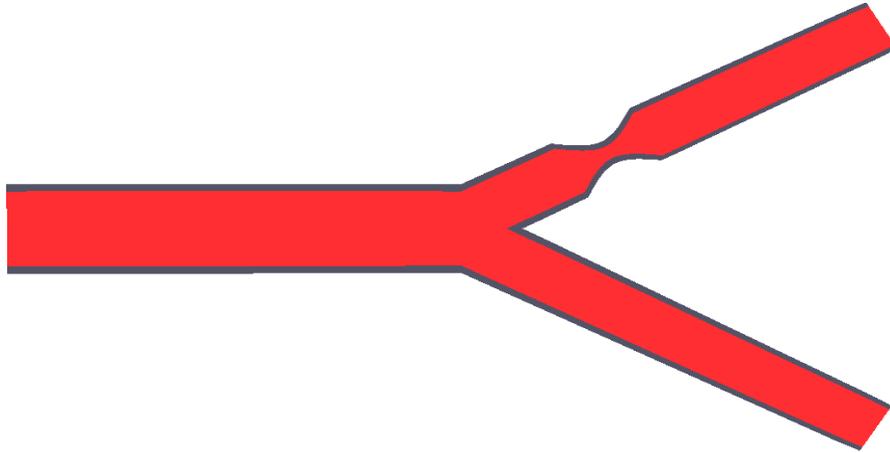


Figure 4.5: Geometric details of the bifurcating artery in 2D. A 75% area reduction stenosis is shown on one of the branches.

cycles with a time step size of 1 ms. Figure 4.6 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 splits nearly in half between the two branches; only a little more flow goes to the 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 splits in a unrealistic pattern, nearly 90% of the flow going to the normal branch. In fact, it has been verified in clinical experiments that the flow distributes almost evenly (50% to 50%) between the normal and stenosed branches, provided that the stenosis is less than 85% in area reduction [82]. In this sense, the simulation obtained with the resistance boundary condition shows a better consistency with the clinical results.

Figure 4.7 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. So far, we have discussed the impact of the resistance and zero-traction outflow boundary condition to the simulation of blood flow in complaint arteries. The resistance boundary condition is shown to be an improvement over the zero-traction outflow boundary con-

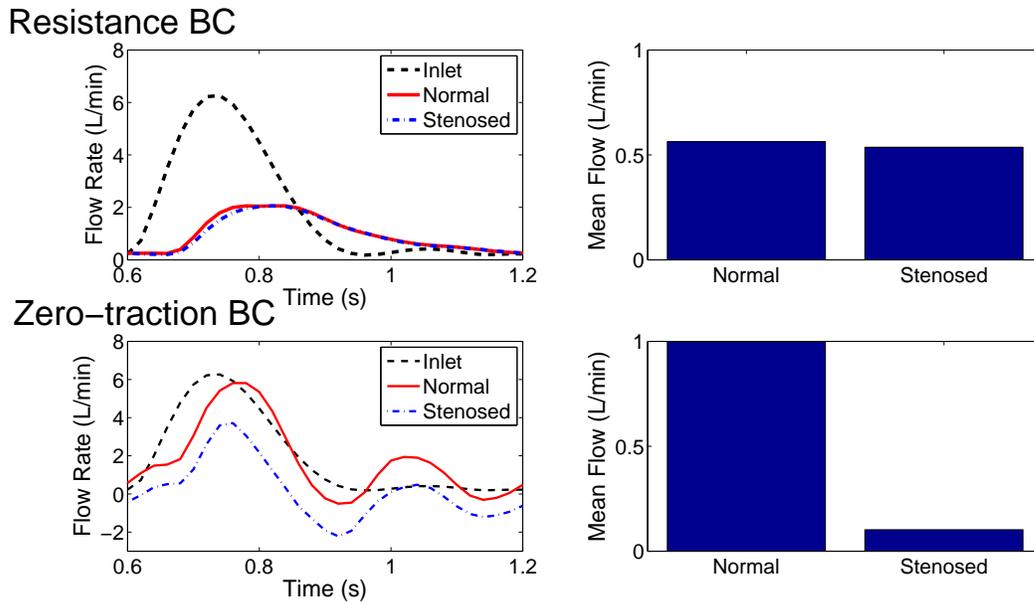


Figure 4.6: Outlet flow waves and flow distribution between the normal artery branch and the stenosed artery branch with a 75% area reduction stenosis during one cardiac cycle, using the resistance and zero-traction outflow boundary condition. Top left plot shows the flow rate at the inlet and outlet boundary during one cardiac cycle 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 using zero-traction boundary conditions are shown in the bottom plots.

dition in obtaining physiological blood flow and pressure.

Figure 4.8 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.

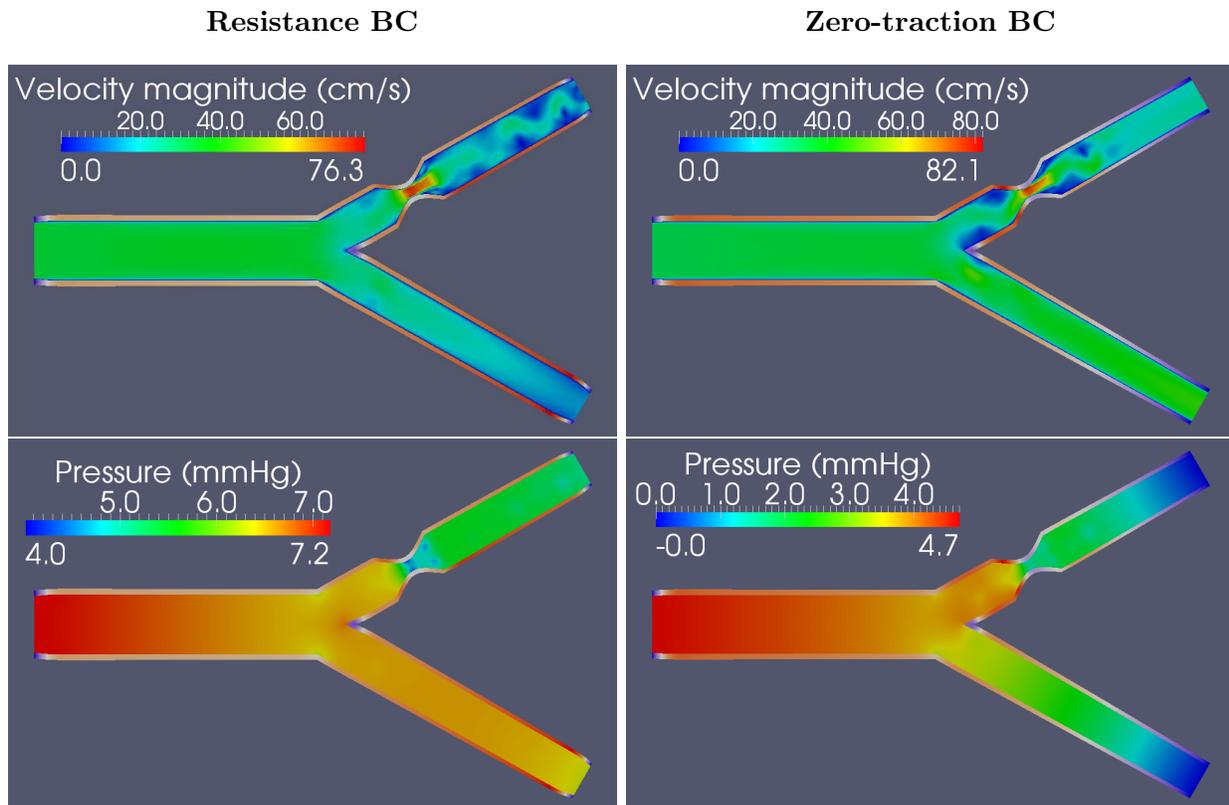


Figure 4.7: Fluid velocity magnitude and pressure at peak systole for the 2D bifurcating artery using resistance and zero-traction outflow boundary conditions. The artery walls are colored by the magnitude of the structural displacement. For the resistance boundary condition, fluid velocity magnitude and pressure are shown on the left. Corresponding figures using the zero-traction boundary are shown on the right.

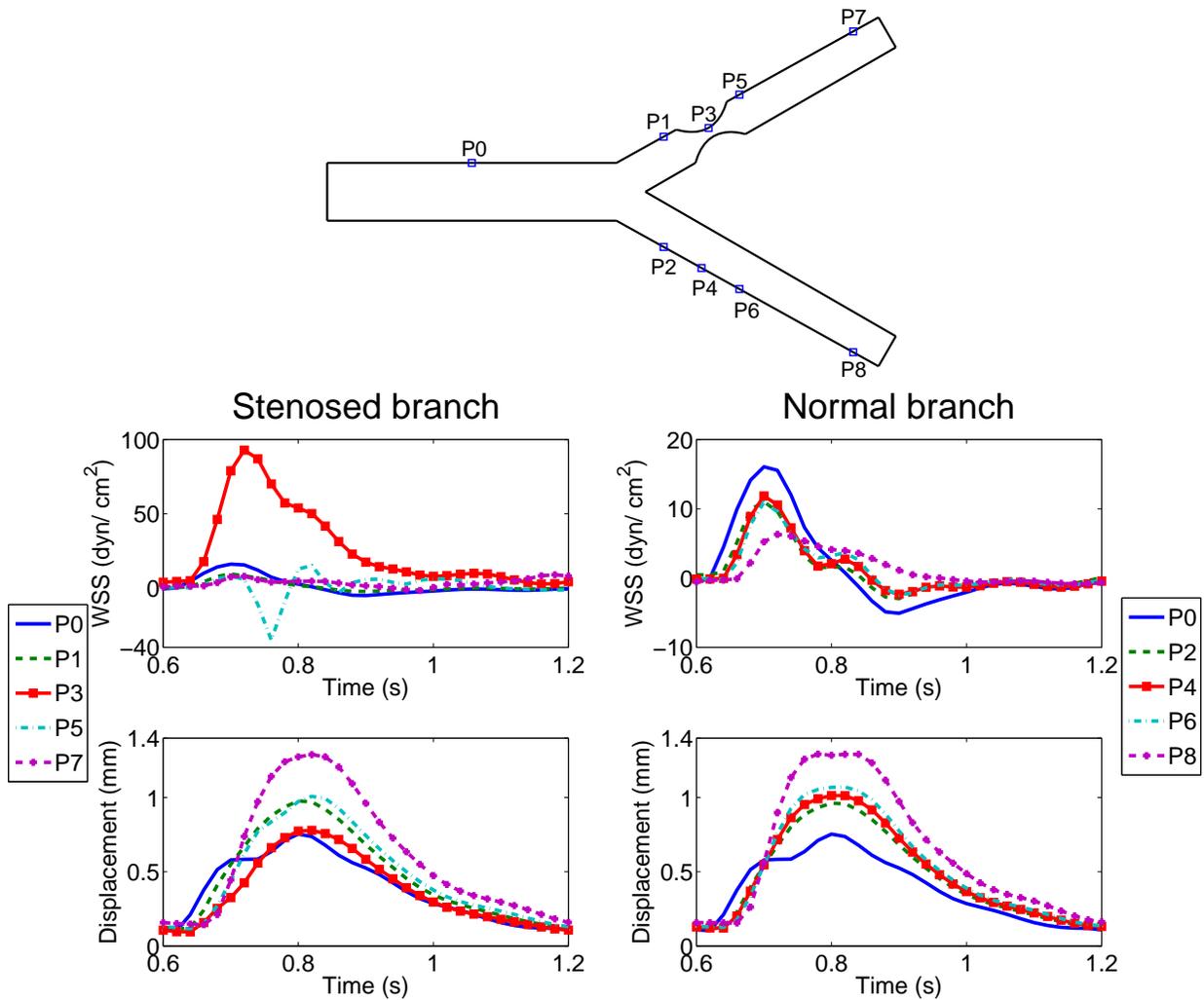


Figure 4.8: Wall shear stress (WSS) and wall displacement at different locations of the artery wall during one cardiac cycle, by using the resistance outflow boundary condition. Top plot shows the locations of measurement on the artery wall. Middle left plot shows the WSS at the locations (P1, P3, P5, 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.

$np$	Zero-traction			Resistance		
	Newton	GMRES	time (s)	Newton	GMRES	time (s)
64	3.0	49.5	300.61	3.1	49.9	316.24
128	3.0	54.3	152.53	3.0	54.7	156.53
256	3.0	87.9	100.52	3.0	90.8	108.50
512	3.0	159.4	54.10	3.0	162.1	54.96

Table 4.1: Performance of the one-level preconditioner with the zero-traction and resistance outflow boundary conditions. The tests are carried on the 2D straight tube problem, associated with the fine mesh of  $2.01 \cdot 10^6$  degrees of freedom. “ $np$ ” denotes the number of processors. “Newton” denotes the average Newton iterations per time step. “GMRES” denotes the average GMRES iterations per Newton step. “Time” refers the average compute time, in seconds, per time step.

## 4.2 Performance and parallel scalability

In this section, we discuss the performance and parallel scalability of the proposed solver to the 2D fully coupled FSI problem using the resistance boundary condition. For all the numerical tests in this section, unless otherwise specified, we use the same geometries, material properties and boundary conditions as described before. The stopping criterion for the Newton iteration is that 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  $\Delta t = 1 \text{ ms}$ , 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. The tests reported in this section are performed on an IBM BlueGene/L with 4096 compute nodes.

We first demonstrate the performance of the one-level (3.2) and cascade two-level (3.8) additive Schwarz preconditioner with the resistance boundary condition, by comparing to the results obtained with the zero-traction boundary condition. In [5, 6], the performance of these preconditioners with the zero-traction boundary condition was studied. As observed in Table 4.1 and 4.2, both the one-level and two-level cascade preconditioners handle the resistance boundary condition well. In terms of the number of iterations and the average compute time, the preconditioner shows

$np$	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 4.2: Performance of the cascade two-level preconditioner with the zero-traction and resistance outflow boundary conditions. The tests are carried on the 2D straight tube problem, associated with the fine mesh of  $2.01 \cdot 10^6$  degrees of freedom and the coarse mesh of  $1.30 \cdot 10^5$  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.

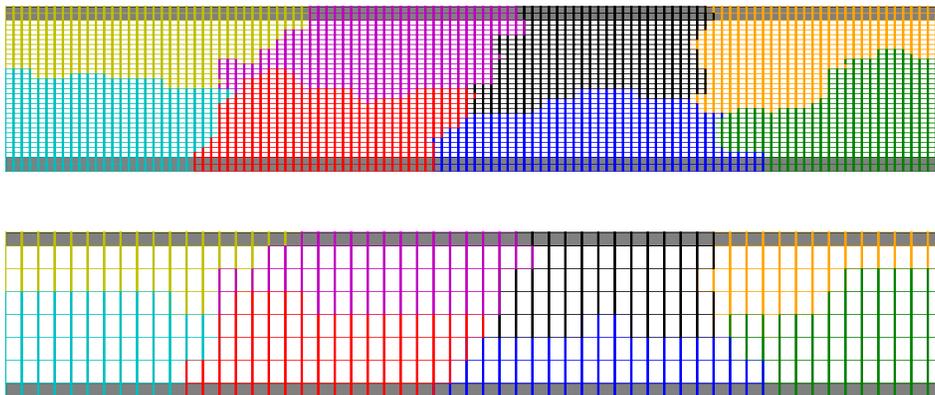


Figure 4.9: Example partitions of the fine mesh and the coarse mesh of a 2D straight tube using ParMETIS. The subdomains are marked with different colors and the structure elements are shaded. The top figure shows the partition of the fine mesh, and the bottom represents a corresponding partition of the coarse mesh. Note that the outlet boundary on the right is shared by two different subdomains.

$np$	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

Table 4.3: Performance of two different preconditioners with increasing number of subdomains for the 2D bifurcating artery problem. The tests are carried on a mesh with  $2.00 \cdot 10^6$  unknowns, using the resistance outflow boundary condition.

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, see Figure 4.9, the one-level and two-level cascade preconditioners still prove to be an effective choice.

In Table 4.3, we show the results obtained with the two-level preconditioner as the number of subdomains increases. Compared to the results of the one-level preconditioner, the performance of the two-level preconditioner is much better. For the one-level additive Schwarz preconditioner, the preconditioned system becomes more ill-conditioned as the number of subdomains increases, while the two-level method results in a very sharp reduction in linear iterations and a good reduction in compute time than the one-level preconditioner.

Our algorithm shows excellent strong scalability as we increase the number of processors into the thousands; see Figure 4.10 for results of the straight tube problem and Figure 4.11 for results of the bifurcating artery problem. 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, which we demonstrate in Figure 4.12. 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. The linear

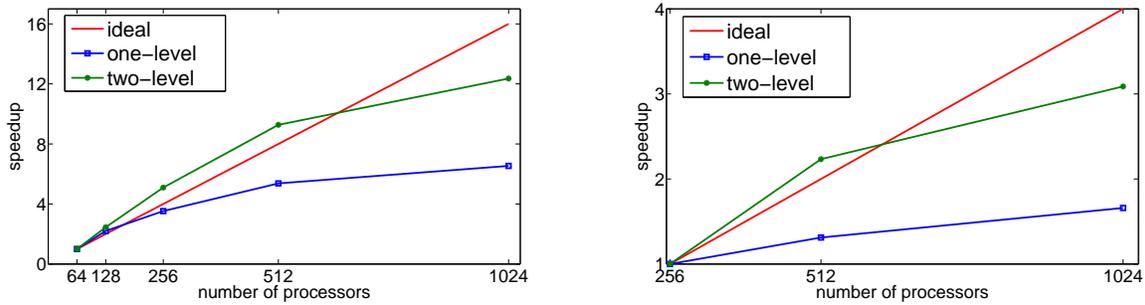


Figure 4.10: Speedup versus number of processors for the 2D straight tube problem with the resistance outflow boundary condition. Results for the problem with 2.01 million unknowns are on the left, and results for the problem with 3.99 million unknowns show on the right.

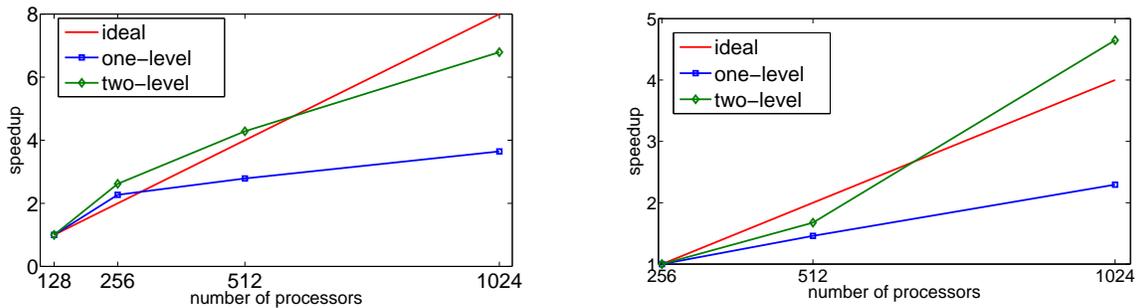


Figure 4.11: Speedup versus number of processors for the 2D bifurcating artery model with the resistance outflow boundary condition. Results for the problem with 2.00 million unknowns is on the left, and results for the problem with 3.88 million unknowns show on the right.

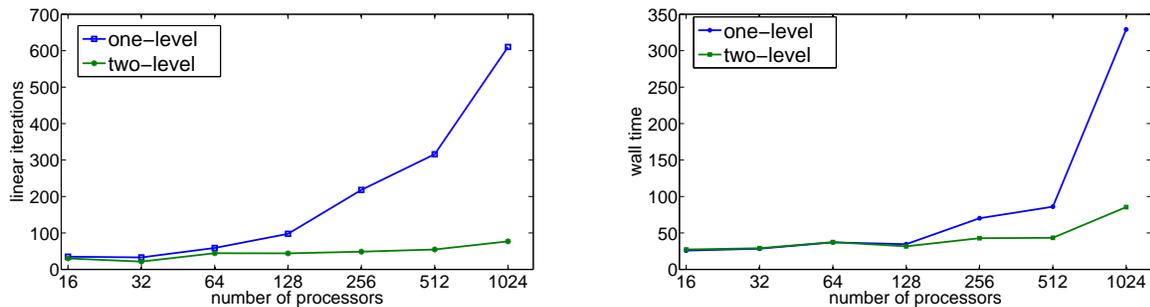


Figure 4.12: Weak scaling of the algorithm for the 2D straight tube problem using the resistance boundary condition. On the left, the vertical axis shows the average number of linear iterations per Newton step. On the right, the vertical axis shows the average compute time in seconds per time step. The number of unknowns increases with the number of processors:  $2.01 \cdot 10^6$  for 256 processors and  $7.99 \cdot 10^6$  for 1024 processors.

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

Table 4.4: The effect of various choices of the preconditioners for the 2D 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	overlap	Newton	fGMRES	time (s)
$9.82 \cdot 10^5$	128	one	1	2.0	132.05	43.28
$9.82 \cdot 10^5$	128	one	2	2.0	83.77	40.77
$9.82 \cdot 10^5$	128	one	3	2.0	60.91	43.42
$9.82 \cdot 10^5$	128	two	0	2.0	69.86	36.41
$9.82 \cdot 10^5$	128	two	1	2.0	65.05	42.12
$9.82 \cdot 10^5$	128	two	2	2.0	49.46	41.50
$2.00 \cdot 10^6$	256	one	1	2.0	471.09	78.16
$2.00 \cdot 10^6$	256	one	2	2.0	317.46	75.94
$2.00 \cdot 10^6$	256	one	4	2.0	177.55	69.80
$2.00 \cdot 10^6$	256	two	0	2.0	51.23	45.29
$2.00 \cdot 10^6$	256	two	1	2.0	86.50	61.13
$2.00 \cdot 10^6$	256	two	2	2.0	128.36	89.07
$3.88 \cdot 10^6$	512	one	4	2.0	219.50	107.15
$3.88 \cdot 10^6$	512	one	6	2.0	149.23	130.29
$3.88 \cdot 10^6$	512	two	0	2.0	114.77	79.87
$3.88 \cdot 10^6$	512	two	1	2.0	120.50	85.72
$3.88 \cdot 10^6$	512	two	2	2.0	95.23	70.67

Table 4.5: The effect of overlapping parameter for the one-level and two-level preconditioners for the 2D bifurcating artery problem with the resistance outflow boundary condition.

iterations are kept very nearly constant for the two-level case in sharp contrast to the one-level preconditioner, and the total compute time behaves similarly.

There are several important factors that impact the performance of the two-level preconditioner, such as the interpolation operator and 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 improving the performance of the preconditioning. In Table 4.4, we show some results obtained with different coarse mesh sizes. As seen, 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.

In the one-level preconditioner, the overlap parameter  $\delta$  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 4.5.

## Chapter 5

### 3D FSI Simulations of blood flows in compliant arteries

In this chapter, we report some numerical results of the proposed fully coupled FSI solver by simulating some blood flows in three-dimensional compliant arteries. Since both the finite element discretization and the solution algorithm, as well as the software, are new, we first validate the correctness of our solver by testing on a well-understood benchmark problem. We then investigate the numerical behavior and parallel performance of our solver with two complex branching geometries derived from clinical data provided by colleagues of the University of Colorado Medical School.

Our solver is implemented on top of the Portable Extensible Toolkit for Scientific computing (PETSc) library [4]. Mesh generations are carried out by CUBIT of Sandia National Laboratories [1] and mesh partitions are obtained with ParMETIS of University of Minnesota [46]. All computations are performed on the Dell PowerEdge C6100 Cluster at the University of Colorado at Boulder.

#### 5.1 Validation

Validation of realistic fluid-structure interaction problem is difficult, because there are no true analytic solutions and few published results. To validate our algorithm, we use a series of test problems. We first test the correctness of our solver for the elastic structure problem and then for the fluid problem with a moving domain. Finally, we validate our FSI solver by testing on a well-understood benchmark problem [20, 25], as shown in the next section.

### 5.1.1 Structure solver

We validate the structure part of our proposed fluid-structure solver using analytic solutions for the linear elasticity equation:

$$\begin{aligned} \rho_s \frac{\partial^2 \mathbf{x}_s}{\partial t^2} - \nabla \cdot \sigma_s &= \mathbf{f}_s \quad \text{in } \Omega_s, \\ \mathbf{x}_s &= g \quad \text{on } \Gamma_s, \end{aligned}$$

where the domain  $\Omega_s \in R^3$  is defined as a pipe with inner radius of 0.5 cm, outer radius of 1.0 cm and length of 1.0 cm. For this problem, we consider the Young's Modulus as  $E = 6.0 \cdot 10^5$  g/(cm s<sup>2</sup>) and the Poisson ratio as  $\nu_s = 0.48$ , and specify the analytic solution as

$$\mathbf{x}_s(x, y, z, t) = \begin{pmatrix} \sin(\pi x) \sin(\pi y) \sin(\pi z) \sin t \\ 2 \sin(\pi x) \sin(\pi y) \sin(\pi z) \sin t \\ \sin(\pi x) \sin(\pi y) \sin(\pi z) \sin t \end{pmatrix}.$$

The homogeneous Dirichlet condition  $\mathbf{x}_s = 0$  is imposed on the inlet boundary, while the body force  $\mathbf{f}_s$  and the Neumann conditions on the other boundaries are the exact solution that is given by the solution above. For the discretization, we discretize the problem with P1 tetrahedral elements in space, and with backward Euler scheme in time. Convergence results for this problem is given in Figure 5.1.

### 5.1.2 Fluid solver in moving domain

We next test our fluid solver in a moving domain using the analytic Womersley solution in a straight circular cylinder. On a given cylinder domain  $\Omega_f^0$  with radius of  $R = 0.24$  cm and length of  $L = 3$  cm, we first define a Womersley flow characterized by the Womersley number  $\alpha = \sqrt{\rho_f \omega / \mu_f} R = 12$ , where  $\omega$  is the angular frequency of the Womersley flow,  $\mu_f = 0.04$  cm/s<sup>2</sup> is the viscosity, and  $\rho_f = 1.0$  g/cm<sup>3</sup> is the fluid density. Based on the derivations in [73, 94], the analytic Womersley solution for the axial velocity  $U_f$  and the pressure  $p_f$  are given by

$$\begin{cases} U_f(x, y, z, t) = \frac{k_s}{4\mu_f} (y^2 + z^2 - R^2) + \text{Im} \left( \frac{ik_\theta}{\rho\omega} \left( 1 - \frac{J_0(\zeta)}{J_0(\Lambda)} \right) e^{i\omega t} \right), \\ p_f(x, y, z, t) = (k_s + k_\theta \sin(\omega t))(x - L), \end{cases}$$

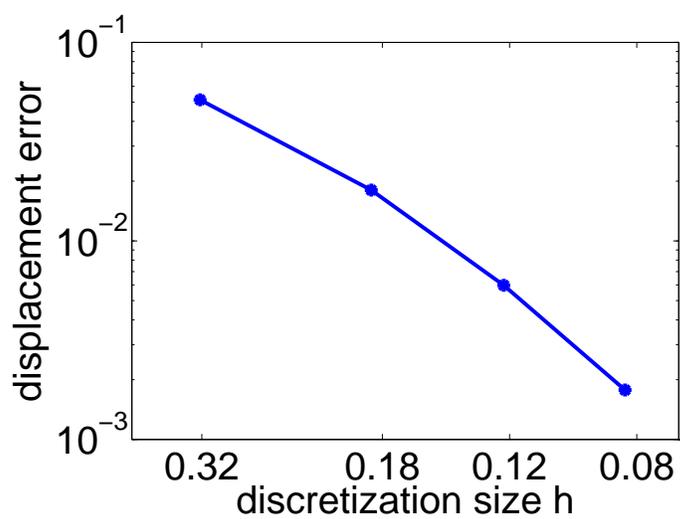


Figure 5.1: Convergence to the analytic solution (at  $t = 0.5s$ ) with respect to the spatial discretization size  $h$  for the 3D structure test problem. The y-axis reports  $L^2$  norm of the displacement error. The temporal discretization is implemented by the backward Euler scheme with the time step size of  $\Delta t = 0.0025s$ .

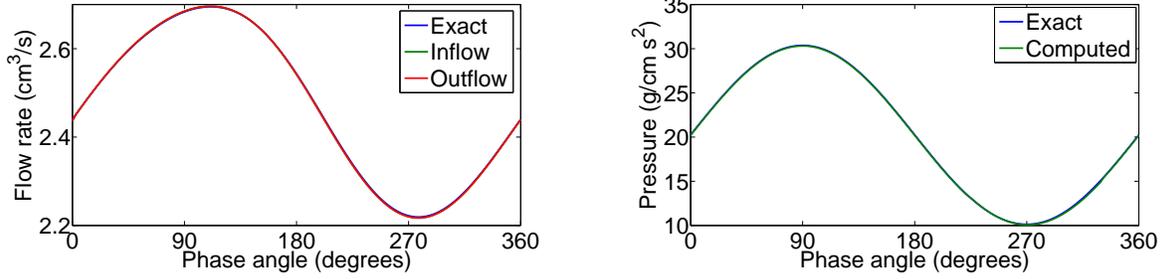


Figure 5.2: Comparisons of the computed solutions with the exact analytic solutions of the 3D fluid problem with moving domain. The tests are carried on a mesh with  $2.17 \cdot 10^6$  unknowns. The left plot shows the flow rate at the inlet and outlet boundary during one cycle, comparing to the exact analytic flow rate. The right plot shows the computed pressure at the center of the cylinder during one cycle, comparing to the exact analytic pressure at the center.

where  $k_s$  and  $k_\theta$  are the steady part and the oscillatory part of the pressure gradient, the parameters  $\Lambda$  and  $\zeta$  are defined as  $\Lambda = \sqrt{-i\alpha}$  and  $\zeta = \Lambda \sqrt{y^2 + z^2}/R$  and  $J_0$  is the Bessel function of order zero of the first kind.

We then define the moving domain  $\Omega_f^t$  as a cylinder of the same length as  $\Omega_f^0$  but with varying radius, where the radius is prescribed by a function of time

$$r(t) = 0.2 + 0.02 \sin(\omega t).$$

Therefore, the moving domain  $\Omega_f^t$  is completely immersed in the given Womersley flow field. For this moving domain problem, we specify the boundary conditions as follows: the analytic inlet pressure is prescribed on the inlet boundary; the stress-free condition is used on the outlet boundary, and the Womersley solution is imposed as the Dirichlet boundary conditions on the moving walls. In this setting, the exact solutions of the moving domain problem can be obtained, which is the restriction of the Womersley solution defined above to the moving domain  $\Omega_f^t$ .

In Figure 5.2, we verify our fluid solver to this moving domain problem by comparing the inflow and outflow flow rate with the analytic flow rate during the first cycle, and by comparing the computed pressure with the analytic solution at the center of the cylinder during the first cycle. As seen, both the computed flow rate and pressure show very good agreement to the analytic solution. In Table 5.1, we report the convergence results with respect to the spatial discretization size  $h$  at

phase angle	h	velocity error	pressure error
90	0.0924	$8.75 \cdot 10^{-3}$	$3.39 \cdot 10^{-3}$
	0.0585	$2.00 \cdot 10^{-3}$	$6.33 \cdot 10^{-4}$
	0.0365	$6.75 \cdot 10^{-4}$	$1.73 \cdot 10^{-4}$
270	0.0924	$8.19 \cdot 10^{-3}$	$6.79 \cdot 10^{-3}$
	0.0585	$2.18 \cdot 10^{-3}$	$1.69 \cdot 10^{-3}$
	0.0365	$8.41 \cdot 10^{-4}$	$4.43 \cdot 10^{-4}$

Table 5.1: Convergence to the analytic solutions at different phases with respect to the discretization size  $h$  for the 3D fluid problem with moving domain.

the phases (phase angle equal to 90 degrees and 270 degrees) where the mesh deformation is the largest.

## 5.2 Benchmark test case

The setup of the benchmark 3D FSI problem consists of a straight cylinder representing the fluid domain with length 5 cm and radius 0.5 cm, and the surrounding wall with thickness 0.1 cm. A constant traction  $\sigma_f \cdot \mathbf{n} = 1.33 \cdot 10^4 \text{ dyn/cm}^2$  is imposed on the inlet boundary for 3 ms. A zero traction condition is applied to the fluid at the outlet boundary. The fluid is characterized with viscosity  $\mu_f = 0.03 \text{ g/(cm s)}$ , and density  $\rho_f = 1.0 \text{ g/cm}^3$ . The Young's modulus  $E = 3 \cdot 10^6 \text{ g/(cm s}^2\text{)}$ , the Poisson ratio  $\nu_s = 0.3$ , and the structure density  $\rho_s = 1.2 \text{ g/cm}^3$  are the parameters of the structure model. The damping parameter  $\alpha$  is set to be zero in this case.

The fluid and the structure are initially at rest and the simulation is run for a total time of 10 ms with a time step size  $\Delta t = 0.1 \text{ ms}$ . The simulation proceeds to the next time step when the relative residual of the nonlinear system is less than  $10^{-6}$ . The stopping criterion for the linear solver is when the preconditioned residual is decreased by a factor of  $10^{-6}$ . To validate the correctness of the coupled spatial and temporal discretization scheme, we run a simulation on a mesh with  $2.41 \cdot 10^6$  elements and  $3.08 \cdot 10^6$  degrees of freedom, and show the computed fluid pressure and the structure deformation at  $t = 2.5, 5.0, 10.0 \text{ ms}$  in Figure 5.3. Our results show excellent agreement with the published results [20, 25]. The pressure wave propagation along the

cylinder is observed. The wall structure deforms in response to the propagation of the pressure pulse, which is a key evidence of the fluid-structure interaction.

We next study the parallel performance and scalability of our fully coupled solver with the one-level Schwarz preconditioner for this benchmark problem. We report the average compute time and the nonlinear iteration count per time step, as well as the average GMRES iterations per Newton step in the experiments, where the documented results are average values over the first 10 time steps. As shown in Figure 5.4, our algorithm shows excellent strong scalability up to thousands of processors. Based on the tests with two different meshes consisting of  $1.25 \cdot 10^6$  and  $3.08 \cdot 10^6$  degrees of freedom, the parallel speedup is shown to be nearly linear with up to 2048 processors. Although there is a mild growth in the number of GMRES iterations, the compute time is almost halved as we double the number of processors.

In the one-level overlapping Schwarz preconditioner, the choice of subdomain solver has a significant impact to the overall performance. In Table 5.2, we show the results obtained using several different subdomain solvers including a point-block LU (BLU) and point-block ILU with  $l = 1, 2$  levels of fill-ins (BILU( $l$ )), and the standard pointwise sparse LU. When LU and BLU are used, the subdomain problem is solved exactly and the number of GMRES iterations is the smallest comparing with inexact subdomain solvers. However, in terms of the total compute time, BLU is always faster, in particular, when the number of processors is relatively small. In sparse matrix factorizations, the fill-in ratio is often used to measure the efficiency of the factorization. It is interesting to note that in all point-block factorizations shown in the table, the fill-in ratio is reduced in comparison with the corresponding pointwise versions, even in the case of BLU. By changing LU to BLU, the compute time is reduced by almost 50% when the number of processors is not large. In the cases of BILU( $l$ ), although there is a mild growth in the number of GMRES iterations, the compute time is further reduced. Comparing with the results using LU factorization, using BILU( $l$ ) as the subdomain solve saves nearly 75% of the compute time when the number of processors is small, and saves over 20% of the compute time when the number of processors is large.

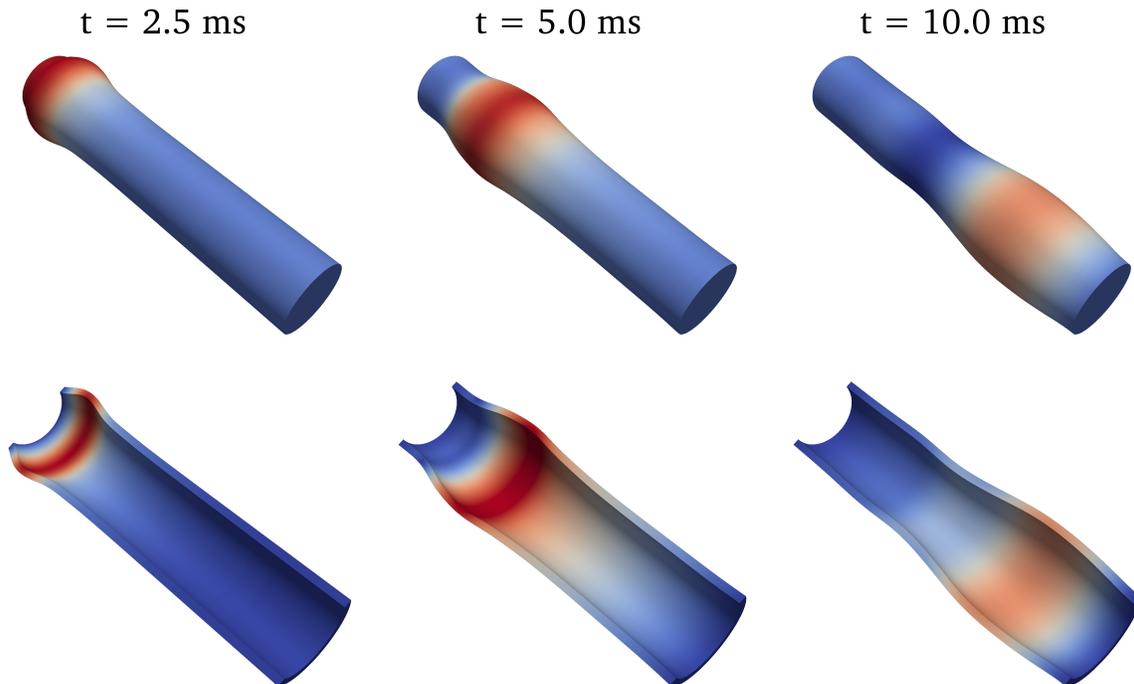


Figure 5.3: Pressure wave propagation (top row) and structure deformation (bottom row) for the 3D straight cylinder case. The simulation is run on a mesh with  $2.41 \cdot 10^6$  elements and  $3.08 \cdot 10^6$  degrees of freedom. The deformation is amplified by a factor of 12 for visualization purpose only.

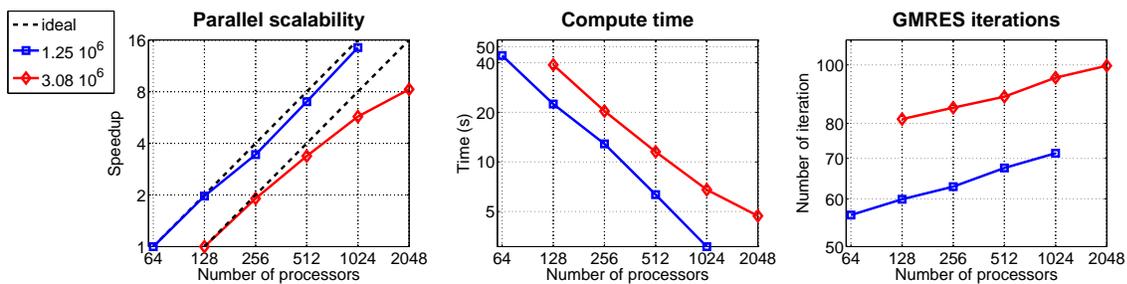


Figure 5.4: Parallel speedup, average compute time per time step, and average number of GMRES iterations per Newton iteration for the 3D benchmark problem with increasing number of processors. The number of unknowns of each problem is listed in the legend. The tests are run with a fixed overlapping size  $\delta = 1$  and BILU(1) is the subdomain solver.

$np$	LU				BLU			
	NI	GMRES	time	fill-in	NI	GMRES	time	fill-in
64	2.0	33.25	218.26	22.84	2.0	33.25	133.47	19.71
128	2.0	39.25	65.21	20.13	2.0	39.25	43.52	14.47
256	2.0	44.25	23.53	14.47	2.0	44.25	16.66	11.35
512	2.0	52.55	10.16	10.38	2.0	52.55	7.56	8.35
1024	2.0	58.85	5.14	7.15	2.0	58.85	3.94	6.71
$np$	BILU(1)				BILU(2)			
	NI	GMRES	time	fill-in	NI	GMRES	time	fill-in
64	2.0	56.40	44.21	2.38	2.0	44.95	38.21	4.79
128	2.0	59.90	22.44	2.36	2.0	48.95	25.73	4.67
256	2.0	62.85	12.07	2.35	2.0	52.55	13.53	4.57
512	2.0	67.50	6.33	2.33	2.0	59.00	7.03	4.47
1024	2.0	71.40	3.06	2.29	2.0	63.35	4.39	4.59

Table 5.2: Performance with respect to the number of processors for different subdomain solvers for the 3D benchmark problem. The tests are carried on a mesh with  $1.25 \cdot 10^6$  unknowns with a fixed overlapping size  $\delta = 1$ . “ $np$ ” denotes the number of processors. “NI” denotes the average number of Newton iterations per time step. “GMRES” denotes the average number of GMRES iterations per Newton step. “time” refers to the average compute time, in seconds, per time step. “fill-in” refers to the average fill-in ratio needed in the subdomain factorization per iteration.

### 5.3 Two-branch artery case

In this section, we report on simulations of a bifurcating artery with two branches that is part of the pulmonary artery of a patient. The artery wall thickness is assumed to be 10% of the local arterial diameter. For the inlet, we prescribe a pulsatile periodic flow wave, with a period  $T$  of 0.6 s. For the outlets, the relation  $P = QR$  is implicitly prescribed on the outflow boundaries as the resistance boundary condition, where  $P$  is the fluid pressure and  $Q = \int_{\Gamma} \mathbf{u}_f \cdot \mathbf{n} \, ds$  represents the flow rate at the outflow boundaries [26, 88]. The resistance is set to  $R = 1408.0 \, \text{dyn} \cdot \text{s}/\text{cm}^5$  and  $R = 677.6 \, \text{dyn} \cdot \text{s}/\text{cm}^5$  at the left and the right artery outlet, respectively. The elastic artery wall is characterized with density of  $1.2 \, \text{g}/\text{cm}^3$ , Young's modulus of  $1.5 \times 10^6 \, \text{g}/(\text{cm} \, \text{s}^2)$ , and Poisson ratio of 0.48. As mentioned earlier, the external force from the surrounding tissues is modeled as a damping term added to the elasticity equation, and the coefficient of the term is chosen as  $\alpha = 6.0 \times 10^3$ , which is the same as what is used in the literature [72, 77]. The blood is modeled with a density of  $1.0 \, \text{g}/\text{cm}^3$  and viscosity of  $0.035 \, \text{g}/(\text{cm} \, \text{s})$ . The geometry and measured resistance values for this model come from clinical data, provided by University of Colorado Medical School. We initialize the Newton iteration by setting the initial wall velocity to zero and using the solution of the steady state FSI problem as the initial condition for the unsteady problem. The simulations are run for 3 cardiac cycles with a time step size of 0.001 s. In Figure 5.5, we show the computed flow speed results and pressure at the inlet, the left artery outlet (LPA), and the right artery outlet (RPA) during one cardiac cycle. The outflow lags the inflow due to the compliance of the artery wall. The computed phase shift between the inflow and the outflow at RPA is 0.02 s. Figure 5.6 shows two snapshots of the fluid velocity field at two phases of the cardiac cycle, the peak systole and early diastole. The fully three-dimensional flow field is quite complex, especially in the diastole phase. Figure 5.7 shows the artery wall velocity vectors at the same two phases, illustrating the corresponding wall movement in response to the fluid dynamics. Such complex flow structures are usually very difficult to be measured clinically; high resolution computation provides a unique way to reveal the phenomena.

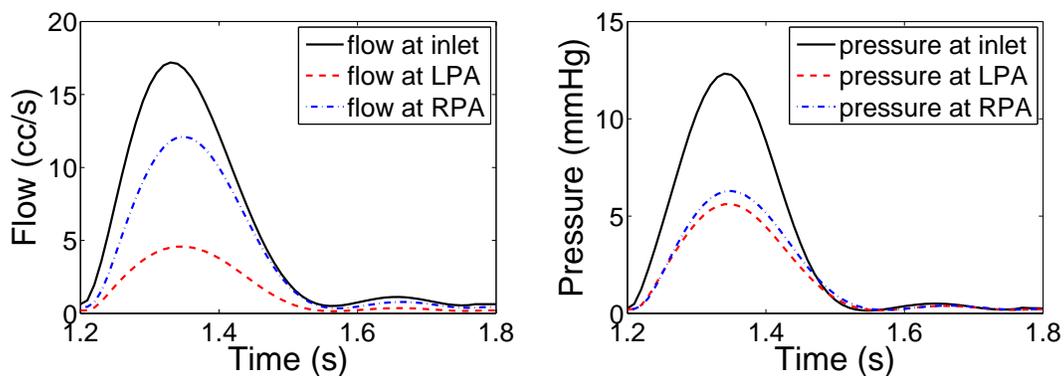


Figure 5.5: Flow and pressure at the inlet and outlets over one cardiac cycle for the two-branch artery model, obtained using the resistance outflow boundary condition. Figure on the left represents the flow rate at the inlet and outlets, and figure on the right shows the fluid pressure at the inlet and outlets. The test is carried on a mesh with  $3.57 \cdot 10^6$  element and  $4.61 \cdot 10^6$  degrees of freedom.

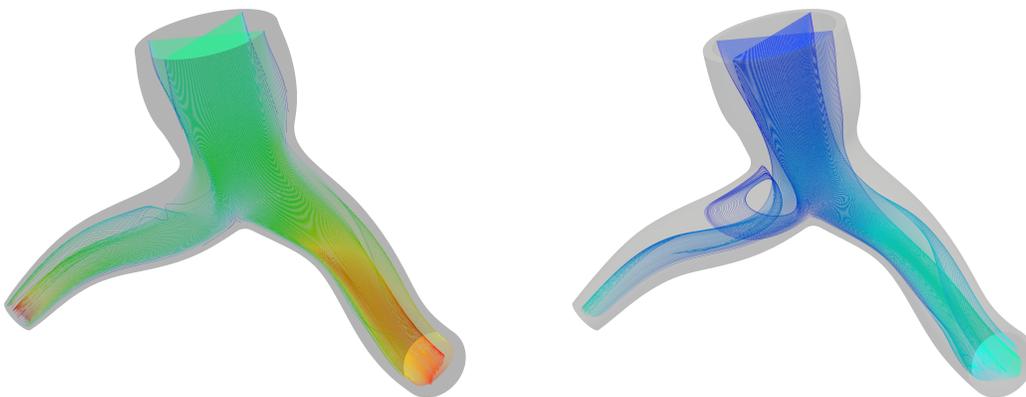


Figure 5.6: Flow in part of the pulmonary artery with two branches at the peak systole (left) and the early diastole (right). The fluid streamlines are colored by velocity magnitude.

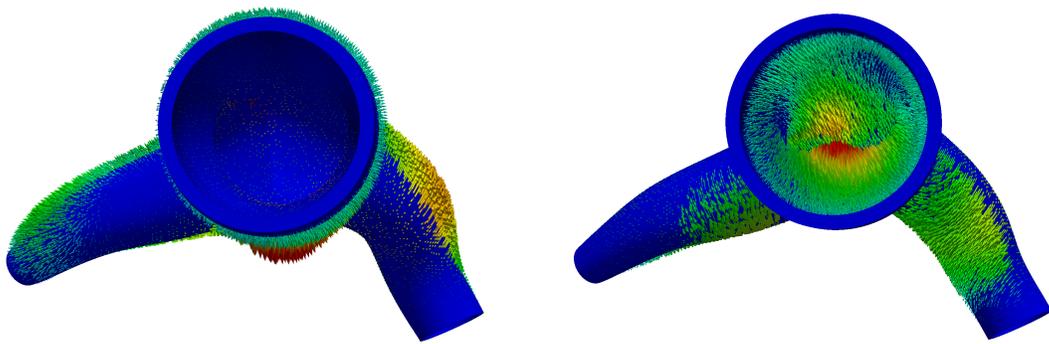


Figure 5.7: Arterial wall velocity vectors obtained at two points of the cardiac cycle: peak systole (left) and early diastole (right).

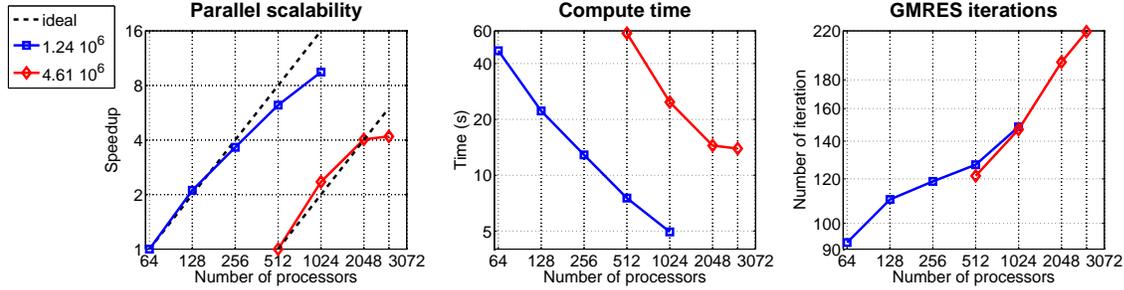


Figure 5.8: Parallel speedup, average compute time per time step, and average number of GMRES iterations per Newton iteration for the two-branch problem with increasing number of processors. The number of unknowns of each mesh is listed in the legend. The tests are run with a fixed overlapping size  $\delta = 1$ .

To investigate the parallel performance and scalability of our one-level solver for this problem, we choose the same linear and nonlinear stopping criteria as in the previous benchmark tests. The time step size is fixed as  $\Delta t = 0.001$  s, and the simulation is stopped after 10 time steps. Figure 5.8 shows the strong scalability for solving the problem discretized on two meshes, one with  $1.24 \cdot 10^6$  degrees of freedom and the other with  $4.61 \cdot 10^6$  degrees of freedom. For the small mesh, the strong scalability stays close to being linear until the number of processors becomes larger than 512. For the larger mesh, the strong scalability stays close to being linear until  $np = 2048$ . It is worth noting that the growth in the number of GMRES iterations for large processor counts may be a problem if we consider solving the problem on a much larger mesh and with a larger number processors. In those situations, one possible solution to improve the scalability is to use a multilevel method. In Table 5.3, we show the results of one-level preconditioner with respect to different subdomain solvers. We should mention that GMRES fails to converge when using BILU(1) as the subdomain solver. But by increasing the fill-in level from 1 to 2 in the point-block incomplete factorization, we are able to reduce the ill-conditioning effect and GMRES converges nicely.

#### 5.4 Complex branching artery case

Next, we conduct experiments for a larger and more complicated artery with many branches obtained from a biplane angiography image of the pulmonary artery of a patient. We assume the

$np$	LU				BLU			
	NI	GMRES	time	fill-in	NI	GMRES	time	fill-in
128	2.0	62.50	95.38	16.73	2.0	62.50	50.73	14.66
256	2.0	76.50	35.16	13.23	2.0	76.50	21.92	10.62
512	2.0	102.25	17.06	8.91	2.0	102.30	9.58	8.91
1024	2.0	129.45	8.09	6.57	2.0	129.45	6.18	5.10
$np$	BILU(2)				BILU(3)			
	NI	GMRES	time	fill-in	NI	GMRES	time	fill-in
128	2.0	110.30	22.21	4.76	2.0	71.85	25.43	8.11
256	2.0	118.75	12.87	4.58	2.0	83.50	14.93	7.87
512	2.0	127.15	7.53	4.75	2.0	111.75	9.08	7.62
1024	2.0	148.50	4.94	4.86	2.0	131.45	5.71	7.84

Table 5.3: Performance with respect to the number of processors for different subdomain solvers for the two-branch model. The tests are carried on a mesh with  $1.24 \cdot 10^6$  unknowns with a fixed overlapping size  $\delta = 1$ . “ $np$ ” denotes the number of processors. “NI” denotes the average number of Newton iterations per time step. “GMRES” denotes the average number of GMRES iterations per Newton step. “time” refers to the average compute time, in seconds, per time step. “fill-in” refers to the average fill-in needed in the factorization per iteration.

wall thickness is 10% of the arterial diameter. The artery density is  $1.2 \text{ g/cm}^3$ ; the Young's modulus of the artery is  $7.5 \times 10^5 \text{ g/(cm s}^2)$ ; and the Poisson ratio is 0.48. The fluid density is  $1.0 \text{ g/cm}^3$  and the viscosity is  $0.035 \text{ g/(cm s)}$ . For the inlet, we prescribe a pulsatile periodic flow with period of  $T = 1.0 \text{ s}$ , mapped to a parabolic velocity profile. The zero-traction boundary conditions are imposed at the outflows. Since the purpose of these simulations is mainly to test the performance of our algorithm, the choice of boundary conditions may not be physiological realistic, but are chosen from the literature [5, 6]. Simulation results for this branching model are shown in Figure 5.9.

Our one-level algorithm again shows very good scalability to this complicated model; see Figure 5.10. The parallel speedup is nearly linear as we increase the number of processors to 3072. For the one-level additive Schwarz preconditioner, the overlapping parameter  $\delta$  is important in accelerating the convergence of GMRES. In Table 5.4, we show the results with various choices of  $\delta$  on different meshes and number of processors. By increasing  $\delta$ , the average number of GMRES iterations decreases. However, smaller overlapping sizes produce better timing results.

## 5.5 Robustness to physical parameters

Besides parallel performance and scalability, another important consideration in the design of discretization schemes and solution algorithms for the fluid-structure interaction problems is the robustness with respect to some of the important physical parameters, in particular, the fluid density and the wall density. It has been reported that the convergence becomes more difficult to achieve if the density of the fluid and the artery wall are close to each other [12], or if the fluid is much denser than the artery wall [57]. Table 5.5 shows that our fully coupled solver with the one-level preconditioner performs quite well for a wide range of fluid and structure densities in terms of the number of Newton iterations, the number of GMRES iterations, and the total compute time.

There are two important physical parameters in describing the properties of the artery wall, the Young's modulus  $E$ , which is related to the stiffness of the artery, and the Poisson ratio  $\nu$ , which represents the incompressibility of the artery. The problems become harder to solve numerically as  $E$  becomes large and  $\nu$  is closer to 0.5. In Table 5.6, our one-level algorithm shows robust

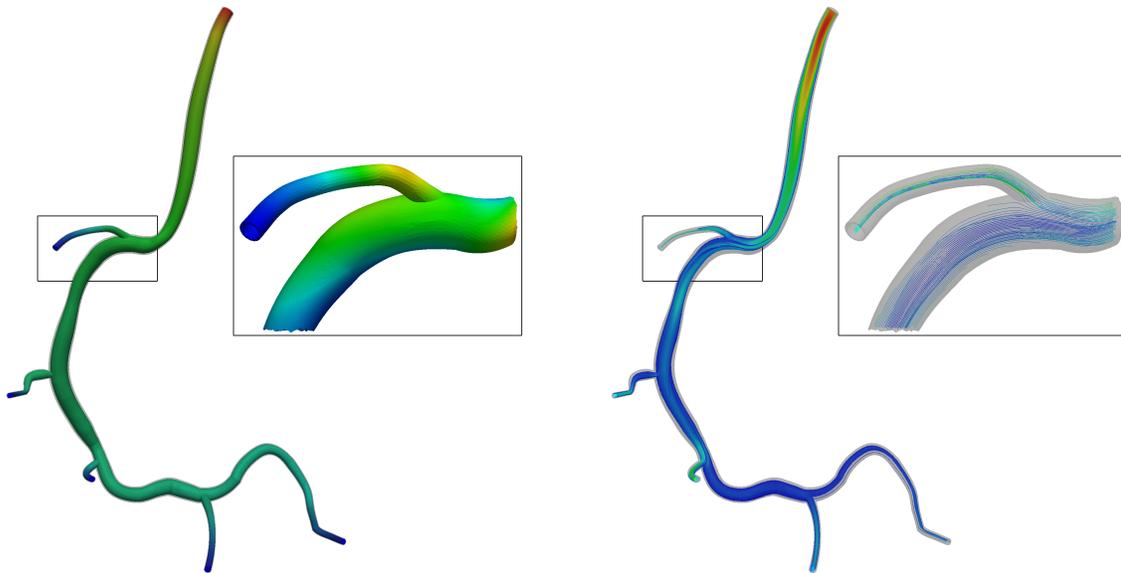


Figure 5.9: Results of the simulation of part of the pulmonary artery. In the large images, the fluid shaded by pressure is shown on the left and the fluid velocity colored in its magnitude is shown on the right. The artery wall is shown in a solid shade in both images. In the inset images, the artery wall shaded by the norm of the displacement is shown on the left and the fluid streamlines colored by vorticity is shown on the right.

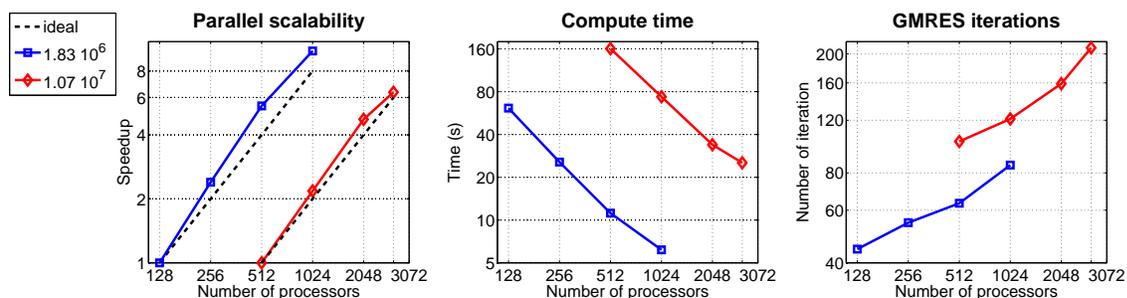


Figure 5.10: Parallel speedup, average compute time per time step, and average GMRES iterations per Newton iteration for the complex branching problem with increasing number of processors. The number of unknowns of each problem is listed in the legend. The tests are run with a fixed overlapping size  $\delta = 1$ .

unknowns	$np$	overlap $\delta$	Newton	GMRES	time
$1.83 \cdot 10^6$	256	1	2.0	54.50	25.58
$1.83 \cdot 10^6$	256	2	2.0	47.45	29.85
$1.83 \cdot 10^6$	256	3	2.0	42.85	33.03
$1.83 \cdot 10^6$	512	1	2.0	63.30	11.89
$1.83 \cdot 10^6$	512	2	2.0	55.80	13.06
$1.83 \cdot 10^6$	512	3	2.0	51.60	16.05
$1.83 \cdot 10^6$	1024	1	2.0	84.95	6.16
$1.83 \cdot 10^6$	1024	2	2.0	71.65	6.61
$1.83 \cdot 10^6$	1024	3	2.0	65.05	9.51
$1.07 \cdot 10^7$	512	1	2.0	101.90	160.50
$1.07 \cdot 10^7$	512	2	2.0	74.15	197.05
$1.07 \cdot 10^7$	512	3	2.0	64.50	258.04
$1.07 \cdot 10^7$	1024	1	2.0	121.20	73.66
$1.07 \cdot 10^7$	1024	2	2.0	96.90	94.40
$1.07 \cdot 10^7$	1024	3	2.0	88.20	140.72
$1.07 \cdot 10^7$	2048	1	2.0	159.00	33.89
$1.07 \cdot 10^7$	2048	2	2.0	118.80	38.19
$1.07 \cdot 10^7$	2048	3	2.0	102.50	54.88

Table 5.4: The effect of various choices of overlapping parameter  $\delta$  on different mesh sizes and number of processors. These tests are for the complex branching problem. “ $np$ ” denotes the number of processors. “Newton” denotes the average Newton iteration per time step. “GMRES” denotes the average GMRES iterations per Newton step. “time” refers to the average compute time, in seconds, per time step.

$\rho_f$	$\rho_s$	Newton	GMRES	time (s)
0.01	1.0	2.0	231.15	9.95
0.1	1.0	2.0	142.65	8.28
1.0	1.2	2.0	85.300	6.164
10.0	1.0	2.1	137.571	8.594
100.0	1.0	3.0	99.267	11.132
1.0	0.01	2.0	87.550	8.809
1.0	0.1	2.0	87.400	9.1795
1.0	10.0	2.0	164.100	11.353
1.0	100.0	2.0	104.600	9.320

Table 5.5: Performance for different combinations of fluid density  $\rho_f$  and wall structure density  $\rho_s$  for the complex branching model. The dynamic viscosity  $\nu_f$  is kept as  $0.035 \text{ cm}^2/\text{s}$ . The tests are run on a mesh with  $1.83 \cdot 10^6$  degrees of freedom and 1024 processors.

convergence with respect to both parameters.

## 5.6 Two-level results

In this section, we investigate the numerical behavior and parallel performance of the two-level hybrid preconditioner (3.4) to the fully coupled FSI system. For all the numerical experiments in this section, we use the same geometries, material properties, and boundary conditions as described before in this chapter. For the solver parameters, we stop the Newton iteration when the relative residual is less than  $10^{-6}$ , and stop the linear solver when the relative residual of the Jacobian system is less than  $10^{-6}$ . For the two-level preconditioner, we consider the coarse linear solver to have converged if the relative residual is less than  $10^{-2}$ , and choose fixed overlapping parameter as  $\delta = 1$  and subdomain solver as point-block LU in both the fine-level and coarse-level preconditioners. For the purpose of comparison, all the results for the one-level method reported in this section are obtained by using the same overlapping parameter  $\delta$  and subdomain solver as in the two-level method.

$E$ ( $g/(cm s^2)$ )	$\nu$	Newton	fGMRES	time (s)
$7.5 \cdot 10^5$	0.45	2.0	74.30	5.84
$1.5 \cdot 10^6$	0.45	2.0	112.65	6.58
$3.8 \cdot 10^6$	0.45	2.0	166.70	7.58
$7.5 \cdot 10^6$	0.45	2.0	220.10	8.52
$7.5 \cdot 10^5$	0.45	2.0	74.30	5.84
$7.5 \cdot 10^5$	0.47	2.0	79.85	5.93
$7.5 \cdot 10^5$	0.48	2.0	84.95	6.02
$7.5 \cdot 10^5$	0.49	2.0	182.10	7.86

Table 5.6: Performance for various values of Young’s modulus  $E_s$  and Poisson ratio  $\nu$  for the complex branching model. The tests are run on a mesh with  $1.83 \cdot 10^6$  degrees of freedom and 1024 processors.

### 5.6.1 Benchmark test case

For the one-level additive Schwarz preconditioner, the preconditioned system becomes more ill-conditioned as the number of subdomains increases. The primary motivation to include a coarse-level in the one-level Schwarz preconditioner is to improve the efficiency and the parallel scalability of the proposed monolithic solver. In Table 5.7, we show the results obtained using the one-level (3.2) and two-level hybrid (3.4) preconditioners. Comparing to the results of the one-level preconditioner, the performance of the two-level preconditioner is much better. For both the small problem with over one million degrees of freedom and the large problem with over three million degrees of freedom, using the two-level preconditioner can greatly reduce the number of linear iterations as well as the total compute time. More importantly, in the case of the two-level method, the number of fGMRES iterations stay close to a small constant as we double the number of processors, while the such numbers are increasing rapidly in the one-level implementation.

Weak scaling is another important measurement of the parallel scalability. In the weak scaling tests, the number of unknowns in the problem increases at the same rate with the number of processors, in order to keep the size of subdomain problems unchanged. As seen in 5.11, the weak scalability for the two-level preconditioner is excellent with up to 2048 processors. The linear iterations are nearly constant in the two-level case, which is in sharp contrast to the one-level

unknowns	$np$	One-level			Two-level		
		Newton	fGMRES	time (s)	Newton	fGMRES	time (s)
$1.25 \cdot 10^6$	128	2.0	39.25	43.25	2.0	10.80	22.95
	256	2.0	44.25	16.66	2.0	11.75	10.46
	512	2.0	55.25	7.56	2.0	13.70	5.52
	1024	2.0	58.85	3.94	2.0	15.00	3.21
$3.08 \cdot 10^6$	128	2.0	46.15	124.71	2.0	13.35	62.99
	256	2.0	53.55	58.27	2.0	15.65	28.59
	512	2.0	60.15	26.75	2.0	17.65	14.60
	1024	2.0	71.25	12.35	2.0	15.60	7.82
	2048	2.0	81.45	6.04	2.0	14.55	4.14

Table 5.7: Performance of the one-level and two-level preconditioners with respect to the increasing number of subdomains for the 3D benchmark problem. The heading “unknowns” reports the problem size, “ $np$ ” denotes the number of processors, “Newton” denotes the average Newton iteration per time step, “GMRES” denotes the average GMRES iterations per Newton step, “time” refers to the average compute time, in seconds, per time step.

preconditioner. The total compute time behaves similarly.

### 5.6.2 Two-branch artery case

Here we study the performance of the two-level preconditioner for a more physically realistic example using a branching artery geometry. For these problems, computations are more challenging because of the complicated geometry and relatively high Reynolds number. As shown in Table 5.8, the difference in the number of linear iterations between one-level and two-level is even more evident. The two-level preconditioner is shown to be much more effective than the one-level preconditioner. Using the two-level preconditioner results in a very sharp reduction in the number of linear iterations and a good reduction in compute time by comparing to the results of one-level preconditioner.

Though the two-level method shows excellent results, one difficulty is that the coarse problem is solved by the same method, as the one-level approach on the fine mesh. The coarse solve could be subject to the same ill-conditioning effect as we increase the number of processors; see Figure 5.12. In most of our simulations, though, the coarse problem is just easy enough to solve, so this is not a major consideration. But as we consider to scale our algorithms to larger problem and larger processor counts, degraded efficiency should be expected. One possible solution is to add more levels to the preconditioner.

As for the weak scalability, again the two-level hybrid preconditioner perform much better than the one-level preconditioner; see Figure 5.13. Compared to the one-level preconditioner, using the two-level preconditioner can greatly reduce the number of linear iterations. On the other hand, in terms of compute time, the two-level preconditioner is also shown more effective especially for large problem and with large number of processors.

### 5.6.3 Parameter selection

Different from the one-level preconditioner, we have a very large number of solver parameters to consider in designing a two-level preconditioner. All of these parameters, such as the linear solver tolerances on the fine- and coarse- levels, the interpolation operator, the overlapping parameters,

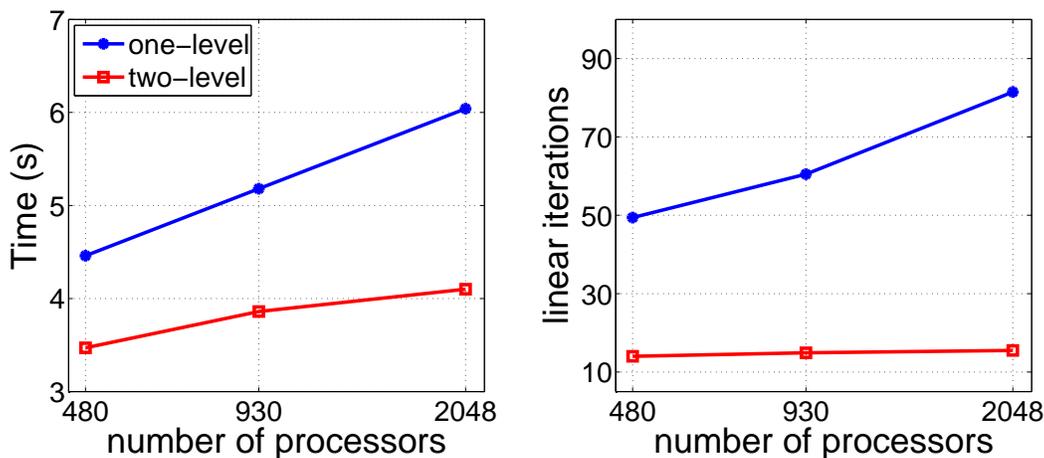


Figure 5.11: Weak scaling of the one-level and two-level preconditioners for the 3D benchmark problem. The reported time and linear iterations are the average compute time per time step and the average GMRES iterations per Newton step, respectively. In these tests, the number of unknowns increases with the number of processors:  $7.18 \cdot 10^5$  for 480 processors,  $1.25 \cdot 10^6$  for 930 processors, and  $3.08 \cdot 10^6$  for 2048 processors.

unknowns	$np$	One-level			Two-level		
		Newton	fGMRES	time (s)	Newton	fGMRES	time (s)
$1.24 \cdot 10^6$	128	2.0	62.50	50.73	2.0	10.25	26.74
	256	2.0	76.50	21.92	2.0	11.20	13.47
	512	2.0	102.30	9.58	2.0	12.20	6.70
	1024	2.0	129.45	6.18	2.0	14.50	4.10
$4.61 \cdot 10^6$	512	2.0	121.45	58.24	2.0	15.50	38.67
	1024	2.0	146.90	24.78	2.0	13.85	18.14
	2048	2.0	193.60	14.45	2.0	18.85	10.69
	3072	2.0	219.25	13.92	2.0	22.10	8.70

Table 5.8: Performance of the one-level and two-level preconditioners with respect to the increasing number of subdomains for the two-branch artery problem.

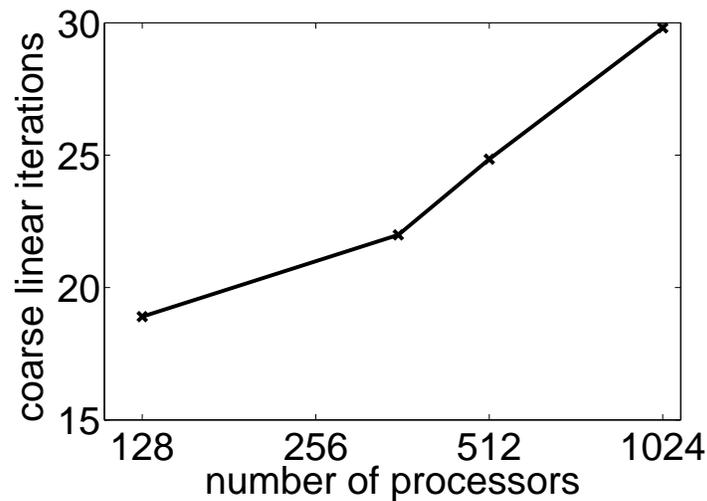


Figure 5.12: The linear iteration counts for the coarse-level solver with respect to the increasing number of subdomains for the two-branch artery problem. This plot shows the average number of linear iterations spent on the coarse-level per fine fGMRES iteration. The tests are carried on a mesh with  $1.24 \cdot 10^6$  unknowns.

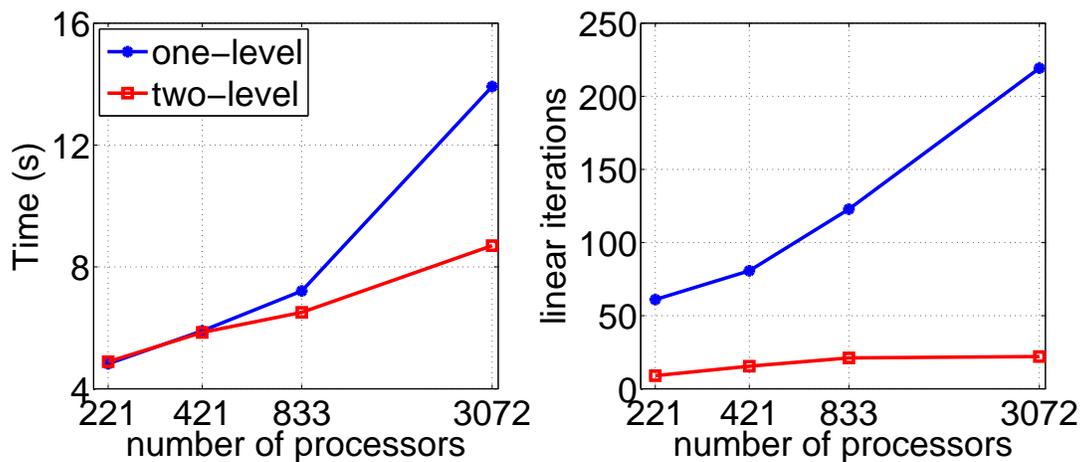


Figure 5.13: Weak scaling of the one-level and two-level preconditioners for the two-branch artery problem. In these tests, the number of unknowns increases with the number of processors:  $3.33 \cdot 10^5$  for 221 processors,  $6.33 \cdot 10^5$  for 421 processors,  $1.24 \cdot 10^6$  for 833 processors, and  $4.61 \cdot 10^6$  for 3072 processors.

unknowns	$np$	coarse size	Newton	fGMRES	time (s)
$1.25 \cdot 10^6$	512	$6.1 \cdot 10^4$	2.0	13.70	5.52
	512	$10.0 \cdot 10^4$	2.0	13.05	5.70
$1.25 \cdot 10^6$	1024	$6.1 \cdot 10^4$	2.0	15.00	3.21
	1024	$10.0 \cdot 10^4$	2.0	15.75	3.34
$3.08 \cdot 10^6$	1024	$6.1 \cdot 10^4$	2.0	17.15	8.41
	1024	$10.0 \cdot 10^4$	2.0	15.65	7.82
	1024	$41.7 \cdot 10^4$	2.0	14.15	9.20
$3.08 \cdot 10^6$	2048	$6.1 \cdot 10^4$	2.0	18.60	5.21
	2048	$10.0 \cdot 10^4$	2.0	14.55	4.14
	2048	$41.7 \cdot 10^4$	2.0	13.20	6.83

Table 5.9: Effect of different coarse mesh sizes on the two-level preconditioner. These results are for the 3D benchmark problem. The heading “coarse size” represents the number of unknowns on the coarse mesh.

unknowns	$np$	coarse size	Newton	fGMRES	time (s)
$1.24 \cdot 10^6$	512	$9.4 \cdot 10^4$	2.0	12.20	6.70
	512	$13.2 \cdot 10^4$	2.0	11.80	7.33
$1.24 \cdot 10^6$	1024	$9.4 \cdot 10^4$	2.0	14.50	4.10
	1024	$13.2 \cdot 10^4$	2.0	12.70	4.23
$4.61 \cdot 10^6$	1024	$33.2 \cdot 10^4$	2.0	13.85	18.14
	1024	$63.3 \cdot 10^4$	2.0	14.75	20.96
$4.61 \cdot 10^6$	2048	$33.2 \cdot 10^4$	2.0	18.85	10.69
	2048	$63.3 \cdot 10^4$	2.0	16.45	11.44

Table 5.10: Effect of different coarse mesh sizes on the two-level preconditioner. These results are for the two-branch artery problem. The heading “coarse size” represents the number of unknowns on the coarse mesh.

unknowns	$np$	coarse rtol	Newton	fGMRES	time (s)	coarse its
$1.25 \cdot 10^6$	256	$10^{-3}$	2.0	11.70	10.95	14.53
	256	$10^{-2}$	2.0	11.75	10.46	9.25
	256	$10^{-1}$	2.0	12.00	9.60	4.68
$1.25 \cdot 10^6$	512	$10^{-3}$	2.0	13.70	6.33	21.81
	512	$10^{-2}$	2.0	13.70	5.52	12.88
	512	$10^{-1}$	2.0	13.75	5.35	5.45
$3.08 \cdot 10^6$	1024	$10^{-3}$	2.0	15.60	8.22	19.98
	1024	$10^{-2}$	2.0	15.60	7.82	12.78
	1024	$10^{-1}$	2.0	16.00	7.85	5.96
$3.08 \cdot 10^6$	2048	$10^{-3}$	2.0	14.50	6.15	27.29
	2048	$10^{-2}$	2.0	14.55	4.14	16.57
	2048	$10^{-1}$	2.0	15.50	4.10	7.49

Table 5.11: Performance of the two-level preconditioner with various parameter values of coarse solver tolerance. These results are for the 3D benchmark problem. The heading “coarse rtol” refers to the relative tolerance for the coarse solve, and “coarse its” denotes the average number of linear iterations spent in the coarse-level solver per fine fGMRES iteration.

and the discretization size on the coarse-level, may have some effect on the overall performance of our proposed method. Therefore, a key question for implementation of the two-level preconditioner is on how to choose the best parameter from the large pool of selections.

Since the motivation for the two-level preconditioner is to improve the performance of the preconditioning by adding a coarse mesh, one of the most important implementation detail is to choose the size of the coarse mesh in order to balance the improvement in conditioning that comes from using a relatively fine coarse mesh with the cost of solving the problem on the coarse mesh. In Table 5.9 and 5.10, we show some results obtained with different coarse mesh sizes. In practice, a relatively fine coarse mesh would help reduce the total number of iterations, but the overall compute time may increase.

Another important parameter, that can have strong effects on the efficiency and parallel scaling of the method, is the tolerance for the coarse level solver. In Table 5.11 and 5.12, we show some results for the two-level preconditioner by varying the solver tolerance on the coarse level. If we choose the coarse solver tolerance as  $10^{-3}$ , the two-level method gives best performance in terms of number of linear iterations. But in terms of the compute time, the best results are obtained with

unknowns	$np$	coarse rtol	Newton	fGMRES	time (s)	coarse its
$1.24 \cdot 10^6$	512	$10^{-3}$	2.0	12.20	7.87	37.55
	512	$10^{-2}$	2.0	12.20	6.70	21.99
	512	$10^{-1}$	2.0	14.70	6.52	11.81
$1.24 \cdot 10^6$	1024	$10^{-3}$	2.0	14.40	5.45	45.31
	1024	$10^{-2}$	2.0	14.50	4.10	24.85
	1024	$10^{-1}$	2.0	17.90	3.69	14.07
$4.61 \cdot 10^6$	1024	$10^{-3}$	2.0	13.85	22.02	70.15
	1024	$10^{-2}$	2.0	13.85	18.14	43.62
	1024	$10^{-1}$	2.0	14.00	14.92	19.90
$4.61 \cdot 10^6$	2048	$10^{-3}$	2.0	18.80	13.08	61.20
	2048	$10^{-2}$	2.0	18.85	10.69	36.51
	2048	$10^{-1}$	2.0	20.15	9.22	14.66

Table 5.12: Performance of the two-level preconditioner with various parameter values of coarse solver tolerance. These results are for the two-branch artery problem. The heading “coarse rtol” refers to the relative tolerance for the coarse solve, and “coarse its” denotes the average number of linear iterations spent in the coarse-level solver per fine fGMRES iteration.

unknowns	$np$	fine sub-solver	Newton	fGMRES	time (s)
$1.25 \cdot 10^6$	256	LU	2.0	11.75	12.19
	256	BLU	2.0	11.75	10.46
	256	BILU(1)	2.0	12.00	8.92
$1.25 \cdot 10^6$	512	LU	2.0	13.70	7.13
	512	BLU	2.0	13.70	5.52
	512	BILU(1)	2.0	13.70	4.61
$3.08 \cdot 10^6$	1024	LU	2.0	15.60	10.02
	1024	BLU	2.0	15.60	7.82
	1024	BILU(1)	2.0	15.10	6.35
$3.08 \cdot 10^6$	2048	LU	2.0	14.55	5.70
	2048	BLU	2.0	14.55	4.14
	2048	BILU(1)	2.0	15.50	3.81

Table 5.13: Performance of the two-level preconditioner with respect to different fine-level subdomain solvers. These results are for the 3D benchmark problem. The headings “fine sub-solver” denotes the choices of subdomain solver for the fine-level preconditioner, where LU stands for the standard LU factorization, BLU for the point-block LU factorization, and BILU(1) for the point-block ILU with one level of fill-ins.

the choice of  $10^{-2}$  or  $10^{-1}$ .

In Table 5.2 and 5.3, we have shown that the choice of the subdomain solver has a significant impact to the performance of the one-level method. For the one-level preconditioner, using the point-block version of incomplete LU factorization can help improving the efficiency in the subdomain solver as well as the overall computation. Same idea can be applied to the two-level preconditioner, we can use the point-block version of incomplete LU factorization to replace the standard LU factorization in the subdomain solver of the fine level preconditioner. Results with respect to different choices of subdomain solvers for the two-level preconditioner are shown in Table 5.13. Although there is a mild growth in terms of number of linear iterations, using BILU(1) as the subdomain solver for the fine-level preconditioner can improve the overall time performance.

## Chapter 6

### Conclusions and future work

#### 6.1 Conclusions

Simulating blood flows in compliant arteries is a challenging problem from both modeling and computational perspectives. On one hand, accurate modeling of blood flows relies on appropriate physical models, effective coupling between the fluid and the structure, and boundary conditions that are used to represent the absent parts of the circulation system. On the other hand, the computation is expensive, therefore the parallel scalability of the solution algorithm becomes a key concern in the simulation. To address these issues, we introduced and studied an ALE based general framework for fully implicit and fully coupled simulation of fluid-structure interaction problems. The development includes the implementation of the resistive outflow boundary conditions, a stabilized finite element discretization on moving unstructured meshes, a parallel monolithic domain decomposition based Newton-Krylov-Schwarz method for solving the coupled system of nonlinear algebraic equations, and a software package runs on supercomputers with thousands of processors.

The major work of the thesis can be divided into two parts. First, we demonstrated the influence of the resistance boundary condition to the accuracy of the solution and discussed the performance of a two-level preconditioner with this integral type resistive boundary condition by solving some two-dimensional test problems. Second, the investigations focus on developing a parallel framework for solving the more physically realistic three-dimensional FSI problems.

Based on the results of many numerical experiments, we demonstrated that the class of one-level and two-level hybrid preconditioners works particularly well for accelerating the convergence

and improving the parallel scalability of our monolithic solver. In particular, the use of a point-block version of ILU in the subdomain preconditioners is found to be beneficial for improving the efficiency of the three-dimensional computation. Moreover, the two-level methods are shown to be effective in removing the dependency of the number of linear iterations on the number of processors, which greatly improves the performance and scalability for larger problems and larger processor counts.

We applied the method to the simulation of blood flows in patient-specific arteries and demonstrated that the algorithm is both accurate and efficient. The algorithm and software show a great deal of robustness with respect to the complicated patient-specific geometries, large meshes, large number of processors, and many important physical parameters. Superlinear scalability was observed for problems with tens of millions of degrees of freedom and on a machine with more than three thousand processors.

## 6.2 Future work

In this thesis, we demonstrated that our algorithm is efficient and effective in simulating blood flows in compliant arteries. The general FSI framework and the parallel scalable solvers in the study can be used for further investigations of the real-world problems with more complicated geometries, and they have the potential to eventually attack a full artery tree simulation in the future. We intend to work on several specific areas in order to improve our models and solution algorithms for the three-dimensional FSI problem:

- Nonlinear structure model. Because the nonlinear, anisotropic behavior of the artery wall structure is important in obtaining better accuracy of the simulation, it is worthwhile to extend our structure problem to include nonlinear visco-elastic equations.
- Reliable outlet boundary conditions. We plan to improve the compatibility of our algorithm with different types of boundary conditions, in order to take advantage of the significant work in the literature on finding more physiologically realistic outlet boundary conditions

for the fluid [48, 83].

- Models for moving domain. The equation we use for modeling the moving domain is simple, and there are possibilities to use some more sophisticated schemes for improved accuracy and robustness.
- Parallel scalability. Though our method scales well with respect to the number of processors in the  $O(10^3)$  range, there are some areas that we need to investigate for improvement. We plan to further extend the approach and the software framework to include additional levels to the preconditioner, which is expected to be more suitable for machines with  $O(10^4)$  processors.

## Bibliography

- [1] The CUBIT Geometry and Mesh Generation Toolkit. Sandia National Laboratories, 2012. <http://cubit.sandia.gov/>.
- [2] S. Badia, F. Nobile, and C. Vergara. Fluid-structure partitioned procedures based on robin transmission conditions. *J. Comput. Phys.*, 227:7027–7051, 2008.
- [3] S. Badia, A. Quaini, and A. Quarteroni. Modular vs. non-modular preconditioners for fluid-structure systems with large added-mass effect. *Comput. Methods Appl. Mech. Engrg.*, 197:4216–4232, 2008.
- [4] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. McInnes, B. Smith, and H. Zhang. PETSc User Manual. Technical report, Argonne National Laboratory, 2012.
- [5] A. T. Barker and X.-C. Cai. Scalable parallel methods for monolithic coupling in fluid-structure interaction with application to blood flow modeling. *J. Comput. Phys.*, 229:642–659, 2010.
- [6] A. T. Barker and X.-C. Cai. Two-level Newton and hybrid Schwarz preconditioners for fluid-structure interaction. *SIAM J. Sci. Comput.*, 32:2395–2417, 2010.
- [7] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and Y. Zhang. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Comput. Mech.*, 43:3–37, 2008.
- [8] Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, and T. E. Tezduyar. 3D simulation of wind turbine rotors at full scale. Part I: Geometry and aerodynamics. *Int. J. Numer. Meth. Fluids*, 65:207–235, 2011.
- [9] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32:199–259, 1982.
- [10] X.-C. Cai. Additive Schwarz algorithms for parabolic convection-diffusion equations. *Numer. Math.*, 60:42–62, 1990.
- [11] X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 21:792–797, 1999.
- [12] P. Causin, J. F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Comput. Methods Appl. Mech. Engrg.*, 194:4506–4627, 2005.

- [13] J. R. Cebal, M. A. Castro, J. E. Burgess, R. S. Pergolizzi, M. J. Sheridan, and C. M. Putman. Characterization of cerebral aneurysms for assessing risk of rupture by using patient-specific computational hemodynamics model. AJNR Am. J. Neuroradiol., 26:2550–2559, 2005.
- [14] A. Codd, T. A. Manteuffel, and S. F. McCormick. Multilevel 1st-order system least squares for nonlinear partial differential equations, with application to elliptic grid generation. SIAM J. Numer. Anal., 41:2197–2209, 2003.
- [15] A. Codd, T. A. Manteuffel, S. F. McCormick, and J. Ruge. Multilevel 1st-order system least squares for elliptic grid generation: ellipticity and computational results. SIAM J. Numer. Anal., 41:2210–2232, 2003.
- [16] R. Codina. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. Comput. Methods Appl. Mech. Engrg., 191:4295–4321, 2002.
- [17] P. Crosetto, S. Deparis, G. Fourestey, and A. Quarteroni. Parallel algorithm for fluid-structure interaction problems in haemodynamics. SIAM J. Sci. Comput., 33:1598–1622, 2011.
- [18] P. Crosetto, P. Raymond, S. Deparis, D. Kontaxakis, N. Stergiopoulos, and A. Quarteroni. Fluid structure interaction simulations of physiological blood flow in the aorta. Computers and Fluids, 43:46–57, 2011.
- [19] J. Dennis and R. Schnabel. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. SIAM, Philadelphia, 1996.
- [20] S. Deparis, M. Discacciati, G. Fourestey, and A. Quarteroni. Fluid-structure algorithms based on Steklov-Poincaré operators. Comput. Methods Appl. Mech. Engrg., 195:5797–5812, 2006.
- [21] S. C. Eisenstat and H. F. Walker. Globally convergent inexact Newton method. SIAM J. Optim., 4:393–422, 1994.
- [22] S. C. Eisenstat and H. F. Walker. Choosing the forcing terms in an inexact Newton method. SIAM J. Sci. Comput., 17:16–32, 1996.
- [23] C. Farhat. CFD on moving grids: from theory to realistic flutter, maneuvering, and multidisciplinary optimization. Int. J. Comput. Fluid Dyn., 19:595–603, 2005.
- [24] C. Farhat, P. Geuzaine, and C. Grandmont. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. J. Comput. Phys., 174:669–694, 2001.
- [25] M. A. Fernandez and M. Moubachir. A Newton method using exact Jacobians for solving fluid-structure coupling. Comput. Struct., 83:127–142, 2005.
- [26] C. A. Figueroa, I. E. Vignon-Clementel, K. E. Jansen, T. J. R. Hughes, and C. A. Taylor. A coupled momentum method for modeling blood flow in three-dimensional deformable arteries. Comput. Methods Mech. Engrg., 195:5685–5706, 2006.
- [27] L. Formaggia, J. F. Gerbeau, F. Nobile, and A. Quarteroni. On the coupling of 3D and 1D Navier-Stokes equations for the flow problems in compliant vessels. Comput. Methods Appl. Mech. Engrg., 191:561–582, 2001.

- [28] L. Formaggia, A. Veneziani, and C. Vergara. A new approach to numerical solution of defective boundary value problems in incompressible fluid dynamics. SIAM J. Numer. Anal., 46:2769–2794, 2008.
- [29] L. P. Franca and S. L. Frey. Stabilized finite element methods: II. The incompressible Navier-Stokes equations. Comput. Methods Appl. Mech. Engrg., 99:209–233, 1992.
- [30] L. Ge and F. Sotiropoulos. A numerical method for solving the 3D unsteady incompressible Navier-Stokes equations in curvilinear domains with complex immersed boundaries. J. Comput. Phys., 225:1782–1809, 2007.
- [31] C. W. Gear. Numerical Initial Value Problems In Ordinary Differential Equations. Prentice-Hall, 1971.
- [32] J. F. Gerbeau and M. Vidrascu. A quasi-Newton algorithm based on a reduced model for fluid structure problems in blood flows. ESAIM - Math. Model. Numer. Anal., 37:631–647, 2003.
- [33] F. J. Gijsen, E. Allanic, F. N. van de Vosse, and J. D. Janssen. The influence of non-Newtonian properties of blood in large arteries: unsteady flow in a 90 degrees curved tube. J. Biomech., 32:705–713, 1999.
- [34] L. Grinberg and G. E. Karniadakis. A scalable domain decomposition method for ultra-parallel arterial flow simulations. Commun. Comput. Phys., 4:1151–1169, 2008.
- [35] D. F. Hawken, J. J. Gottlieb, and J. S. Hansen. Review of some adaptive node-movement techniques in finite-element and finite-difference solutions of partial differential equations. J. Comput. Phys., 95:254–302, 1991.
- [36] M. Heil. An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems. Comput. Methods Appl. Mech. Engrg., 193:1–23, 2004.
- [37] J. J. Heys, C. G. DeGro, T. A. Manteuffel, S. F. McCormick, and H. Tufo. Modeling 3-D compliant blood flow with FOSLS. Biomed. Sci. Instrum., 40:193–199, 2004.
- [38] J. J. Heys, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge. First-order system least squares (FOSLS) for coupled fluid-elastic problems. J. Comput. Phys., 195:560–575, 2004.
- [39] G. A. Holzapfel and T. C. Gasser. A new constitutive framework for arterial wall mechanics and a comparative study of material models. Journal of Elasticity, 61:1–40, 2000.
- [40] M. A. Howard. Finite Element Modeling and Optimization of High-Speed Aerothermoelastic Systems. PhD thesis, University of Colorado at Boulder, 2010.
- [41] T. J. R. Hughes, L. P. Franca, and G. M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin-least-squares method for advective-diffusive equations. Comput. Methods Appl. Mech. Engrg., 73:173–189, 1989.
- [42] T. J. R. Hughes, W. K. Liu, and T. K. Zimmermann. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. Comput. Methods Appl. Mech. Engrg., 29:329–349, 1981.

- [43] F.-N. Hwang and X.-C. Cai. A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible navier-stokes equations. J. Comput. Phys., 204:666–691, 2005.
- [44] F.-N. Hwang and X.-C. Cai. Parallel fully coupled Schwarz preconditioners for saddle-point problems. Electron. Trans. Numer. Anal., 22:146–162, 2006.
- [45] A. A. Johnson and T. E. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. Comput. Methods Appl. Mech. Engrg., 119:73–94, 1994.
- [46] G. Karypis. METIS/ParMETIS web page. University of Minnesota, 2012. <http://glaros.dtc.umn.edu/gkhome/views/metis>.
- [47] O. Kayser-Herold and H. Matthies. A unified least-squares formulation for fluid-structure interaction problems. Comput. Struct., 85:998–1011, 2007.
- [48] H. J. Kim, C. A. Figueroa, T. J. R. Hughes, K. E. Jansen, and C. A. Taylor. Augmented Lagrangian method for constraining the shape of velocity profiles at outlet boundaries for three-dimensional finite element simulations of blood flow. Comput. Methods Appl. Mech. Engrg., 198:3551–3566, 2009.
- [49] A. Klawonn and L. F. Pavarino. Overlapping Schwarz methods for mixed linear elasticity and Stokes problems. Comput. Methods Appl. Mech. Engrg., 165:233–245, 1998.
- [50] D. Knoll and D. E. Keyes. Jacobian-free Newton-Krylov methods: A survey of approaches and applications. J. Comput. Phys., 193:357–397, 2004.
- [51] E. Kuhl, S. Hulshoff, and R. de Borst. An arbitrary Lagrangian Eulerian finite-element approach for fluid-structure interaction phenomena. Int. J. Numer. Methods Eng., 57:117–142, 2003.
- [52] U. Küttler, M. Gee, Ch. Forster, A. Comerford, and W. A. Wall. Coupling strategies for biomedical fluid-structure interaction problems. Int. J. Numer. Biomed. Engng., 26:305–321, 2010.
- [53] U. Küttler and W. A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. Comput. Mech., 43:61–72, 2008.
- [54] J. F. LaDisa, L. E. Olson, R. C. Molthen, D. A. Hettrick, P. F. Pratt, M. D. Hardel, J. R. Kersten, D. C. Warltier, and P. S. Pagel. Alterations in wall shear stress predict sites of neointimal hyperplasia after stent implantation in rabbit iliac arteries. Am. J. Physiol. Heart Circ. Physiol., 288:H2465–H2475, 2005.
- [55] P. T. Lin, J. N. Shadid, M. Sala, R. S. Tuminaro, G. L. Hennigan, and R. J. Hoekstra. Performance of a parallel algebraic multilevel preconditioner for stabilized finite element semiconductor device modeling. J. Comput. Phys., 228:6250–6267, 2009.
- [56] M. Manguoglu, K. Takizawa, S. Tanaka, and T. E. Tezduyar. Numerical-performance studies for the stabilized space-time computation of wind-turbine rotor aerodynamics. Comput. Mech., 48:293–306, 2011.

- [57] C. Michler, E. H. van Brunmmelen, and R. de Borst. The relevance of conservation for stability and accuracy of numerical methods for fluid-structure interactions. Comput. Methods Appl. Mech. Engrg., 192:4195–4215, 2003.
- [58] F. Nobile. Numerical Approximation of Fluid-Structure Interaction Problem with Application to Hemodynamics. PhD thesis, Ecole Polytechnique Federale de Lausanne, 2001.
- [59] M. S. Olufsen. Structured tree outflow condition for blood flow in larger systemic arteries. Am. J. Physiol., 276:H257–H268, 1999.
- [60] M. S. Olufsen, C. S. Peskin, W. Y. Kim, E. M. Pederson, A. Nadim, and J. Larsen. Numerical simulation and experimental validation of blood flow in arteries with structure-tree outflow conditions. Ann. Biomed. Eng., 28:1281–1299, 2000.
- [61] J. T. Ottesen, M. S. Olufsen, and J. K. Larsen. Applied Mathematical Models in Human Physiology. SIAM, Philadelphia, 2004.
- [62] K. Perktold, M. Resch, and R. O. Peter. Three-dimensional numerical analysis of pulsatile flow and wall shear stress in the carotid artery bifurcation. J. Biomech., 24:409–420, 1991.
- [63] F. Petrini, F. Giuliano, and F. Bontempi. Comparison of time domain techniques for the evaluation of the response and the stability in long span suspension bridges. Comput. Struct., 85:1032–1048, 2007.
- [64] A. Quarteroni, M. Tuveri, and A. Veneziani. Computational vascular fluid dynamics: problems, models and methods. Comput. Visual Sci., 2:163–197, 2000.
- [65] Y. Saad. A flexible inner-outer preconditioned gmres algorithm. SIAM J. Sci. Comput., 14:461–469, 1993.
- [66] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear system. SIAM J. Sci. Stat. Comp., 7:856–869, 1986.
- [67] L. Salvatori and C. Borri. Frequency- and time-domain methods for the numerical modeling of full bridge aeroelasticity. Comput. Struct., 85:675–687, 2007.
- [68] B. Smith, P. Bjorstad, and W. Gropp. Domain Decomposition. Cambridge, 1996.
- [69] D. D. Soerensen, K. Pekkan, D. de Zelicourt, S. Sharma, K. Kanter, M. Fogel, and A. P. Yoganathan. Introduction of new optimized total cavopulmonary connection. Ann. Thorac. Surg., 83:2182–2190, 2007.
- [70] R. Spilker, J. Feinstein, D. Parker, V. Reddy, and C. Taylor. Morphometry-based impedance boundary conditions for patient-specific modeling of blood flow in pulmonary arteries. Ann. Biomed. Eng., 35:546–559, 2007.
- [71] G. R. Stuhne and D. A. Steinman. Finite-element modeling of the hemodynamics of stented aneurysms. J. Biomech. Eng., 126:382–387, 2004.
- [72] K. Takizawa, J. Christopher, T. E. Tezduyar, and S. Sathe. Space-time finite element computation of arterial fluid-structure interactions with patient-specific data. Int. J. Numer. Meth. Biomed. Engrg., 26:101–116, 2010.

- [73] C. A. Taylor. A Computational Framework for Investigating Hemodynamic Factor in Vascular Adaptation and Disease. PhD thesis, Stanford University, August 1996.
- [74] C. A. Taylor and M. T. Draney. Experimental and computational methods in cardiovascular fluid mechanics. Ann. Rev. Fluid Mech., 36:197–231, 2004.
- [75] C. A. Taylor, T. J. R. Hughes, and C. K. Zarins. Finite element modeling of blood flow in arteries. Comput. Methods Mech. Engrg., 158:155–196, 1998.
- [76] C. A. Taylor and J. D. Humphrey. Open problems in computational vascular biomechanics: Hemodynamics and arterial wall mechanics. Comput. Methods Appl. Mech. Engrg., 198:3514–3523, 2009.
- [77] T. E. Tezduyar, S. Sathe, T. Cragin, B. Nanna, B. S. Conklin, J. Pausewang, and M. Schwaab. Modelling of fluid-structure interactions with the space-time finite elements: Arterial fluid mechanics. Int. J. Numer. Meth. Fluids, 54:901–922, 2007.
- [78] A. Toseli and O. Widlund. Domain Decomposition Methods—Algorithms and Theory. Springer-Verlag, Berlin, 2005.
- [79] S. A. Urquiza, P. J. Blanco, M. J. Venere, and R. A. Feijoo. Multi-dimensional modelling for the carotid artery blood flow. Comput. Methods Appl. Mech. Engrg., 195:4002–4017, 2006.
- [80] R. van Loon, P. Anderson, F. van de Vosse, and S. Sherwin. Comparison of various fluid-structure methods for deformable bodies. Comput. Struct., 85:833–843, 2007.
- [81] I. Vignon and C. A. Taylor. Outflow boundary conditions for one-dimensional finite element modeling of blood flow and pressure waves in arteries. Wave Motion, 39:361–374, 2004.
- [82] I. E. Vignon-Clementel, C. A. Figueroa, K. E. Jansen, and C. A. Taylor. Outflow boundary conditions for three-dimensional finite element modeling of blood flow and pressure in arteries. Comput. Methods Appl. Mech. Engrg., 195:3776–3796, 2006.
- [83] I. E. Vignon-Clementel, C. A. Figueroa, K. E. Jansen, and C. A. Taylor. Outflow boundary conditions for 3D simulations of non-periodic blood flow and pressure fields in deformable arteries. Comput. Methods Biomech. Biomed. Engin., 13:5625–5640, 2010.
- [84] H. Wang, J. Chessa, W. K. Liu, and T. Belytschko. The immersed/fictitious element method for fluid-structure interaction: volumetric consistency, compressibility and thin members. Int. J. Numer. Meth. Engrg., 74:32–55, 2008.
- [85] C. H. Whiting. Stabilized Finite Element Methods for Fluid Dynamics Using a Hierarchical Basis. PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, 1999.
- [86] C. H. Whiting and K. E. Jansen. A stabilized finite element method for the incompressible Navier-Stokes equations using a hierarchical basis. Int. J. Numer. Meth. Fluids, 35:93–116, 2001.
- [87] Yunhai Wu, X.-C. Cai, and D. E. Keyes. Additive Schwarz methods for hyperbolic equations. Contemp. Math., 218:513–521, 1998.
- [88] Yuqi Wu and X.-C. Cai. A parallel two-level method for simulating blood flows in branching arteries with the resistive boundary condition. Computers and Fluids, 45:92–102, 2011.

- [89] Yuqi Wu and X.-C. Cai. A parallel domain decomposition algorithm for simulating blood flow with incompressible Navier-Stokes equations with resistive boundary condition. Commun. Comput. Phys., (11):1279–1299, 2012.
- [90] Yuqi Wu and X.-C. Cai. A fully implicit domain decomposition based ALE framework for three-dimensional fluid-structure interaction with application in blood flow computation. submitted.
- [91] Yuqi Wu and X.-C. Cai. A parallel monolithic domain decomposition method for blood flow simulations in 3D. submitted.
- [92] C. Yang and X.-C. Cai. Parallel multilevel methods for implicit solution of shallow water equations with nonsmooth topography on the cubed-sphere. J. Comput. Phys., 230:2523–2539, 2011.
- [93] C. Yang, J. Cao, and X.-C. Cai. A fully implicit domain decomposition algorithm for shallow water equations on the cubed-sphere. SIAM J. Sci. Comput., 32:418–438, 2010.
- [94] M. Zamir. The Physics of Pulsatile Flow. Springer-Verlag, 2000.
- [95] Y. Zhang, M. L. Dunn, M. S. Drexler, C. N. McCowen, A. J. Slifka, D. D. Ivy, and R. Shandas. A microstructural hyperelastic model of pulmonary arteries under normo- and hypertensive conditions. Annals of Biomedical Engineering, 33:1042–1052, 2005.
- [96] M. Zhou, O. Sahni, H. J. Kim, C. A. Figueroa, C. A. Taylor, M. S. Shephard, and K. E. Jansen. Cardiovascular flow simulation at extreme scale. Comput. Mech., 46:71–82, 2010.