An Investigation of the Effectiveness of Block Preconditioners on a Non-Newtonian Blood Flow Model

by

Ashlee Renee Fuchs, B.S.

A Thesis

In

Mathematics and Statistics

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for the Degree of

Master of Science

Approved

Victoria E. Howle
Committee Chair

Robert C. Kirby

Kevin R. Long

Peggy Gordon Miller
Dean of the Graduate School

May, 2011

©2011, Ashlee Renee Fuchs

ACKNOWLEDGEMENTS

I would like to extend my gratitude to several people. First, I would like to thank God for blessing me with an amazing support system as well as the intelligence and perseverance needed to pursue and obtain a Master's degree. I want to thank my husband, Derrick, who has always believed in me and has been extremely supportive of my graduate education. I am also very thankful for my family and friends who have encouraged me and cheered me on throughout this thesis project. A special thank you goes to my parents and grandparents for teaching me to strive for excellence and for instilling in me a strong work ethic. I would also like to thank my committee members, Dr. Kevin Long and Dr. Robert C. Kirby, for their input and suggestions. And last, but certainly not least, I would like to thank my advisor and committee chair, Dr. Victoria E. Howle for her guidance and expertise throughout this research project. She has been an excellent mentor and has taught me many valuable lessons not only in the realm of mathematics but also in the realm of life.

CONTENTS

Acknowledgements ii

Abstract v

List of Tables vi

List of Figures vii

1. Introduction 1

2. Navier-Stokes Equations 3

2.1 Newtonian Flow 3

2.2 Non-Newtonian Flow 4

2.3 Model of Channel and Channel with Valve 8

3. Linearization and Discretization 10

3.1 Linearization 10

3.2 Discretization 12

4. Approximating each Subsolve 19

4.1 Iterative Methods 20

4.1.1 GMRES 21

4.2 Preconditioning 21

4.2.1 ILU Preconditioner 22

4.2.2 PCD Preconditioner 25

5. Application of Preconditioners to Flow Through a Channel Problem . . 28

5.1 Trilinos and Sundance 28

5.2 Procedure 29

5.2.1 Step 1: Newtonian flow with direct linear solver 29

5.2.2 Step 2: Newtonian flow with preconditioned iterative solver 30

5.2.3 Step 3: Non-Newtonian flow with preconditioned iterative solver 33

6. Results from Investigation 34

7. Concluding Remarks 46

Bibliography 48

ABSTRACT

Newtonian and non-Newtonian fluid flow can be modeled by the steady-state incompressible Navier-Stokes equations. Developing a non-Newtonian blood flow model is the particular problem of interest for this thesis. The process of solving this problem involves utilizing iterative solvers which can perform with higher accuracy and efficiency if properly preconditioned. Physics-based preconditioners have been developed for solving the steady-state incompressible Navier-Stokes equations. The primary aim of this thesis is to apply the pressure convection-diffusion (PCD) preconditioner to a non-Newtonian flow model in order to investigate effectiveness.

LIST OF TABLES

5.1 *Size, n, corresponding to each of the meshes is reported. Channel refers to the simple channel model (see Figure 2.1) while valve refers to the model of the channel with an obstruction (see Figure 2.2).* 32

6.1 *Results from Newtonian flow model with Oseen linearization and LU factorization applied to linear subproblems at various viscosities for different mesh sizes.* 34

6.2 *Results from Newtonian flow model with preconditioned GMRES at various viscosities for different mesh sizes. Avg GMRES is the average of the number of GMRES iterations at each Oseen step. N indicates that GMRES did not converge within 500 iterations.* 37

6.3 *Results from the non-Newtonian flow model on the channel with preconditioned GMRES at various viscosities for different mesh sizes. Avg GMRES is the average of the number of GMRES iterations at each Oseen step.* 40

6.4 *Results from the non-Newtonian flow model on the valve with preconditioned GMRES at various viscosities for different mesh sizes. Avg GMRES is the average of the number of GMRES iterations at each Oseen step.* 43

LIST OF FIGURES

2.1 *A model of pressure driven flow through a channel with* $\boldsymbol{u}$*=0 along the top and bottom boundary,* $p = 1$ *at the left boundary, and* $p = 0$ *at the right boundary.* 8

2.2 *A simple model of pressure driven flow through a channel with fixed top and bottom walls and a fixed obstruction (valve). For regions 3,4,5,6,7,8,* $\mathbf{u} = \mathbf{0}$*. For region 1,* $p = 1$*. For region 2,* $p = 0$*.* 9

3.1 *Sample meshes: portion of coarsest mesh is pictured on the left and a portion of the next coarsest mesh is on the right* 13

4.1 *Sparsity pattern of the n x n block matrix from the coarsest valve mesh used in this research. Nonzero entries are shown in blue; all other entries are zero. For this matrix,* $n = 12,373$*, and the matrix has 193,341 nonzeros denoted nz in the figure which is* $O(n)$*.* 23

4.2 *Sparsity pattern of the L and U factors from an LU decomposition of the matrix in figure 4.1. For both L and U, the matrix size is* $n = 12,373$*, but each factor has approximately 23 million nonzeros, denoted nz in the figure, which is* $O(n^2)$ *rather than the* $O(n)$ *number of nonzeros in the original matrix.* 24

6.1 *Color map of velocity in the x direction for the coarsest channel without a valve, channel-1, with viscosity* $\nu = 1$*.* 35

6.2 *Color map of velocity in the x direction for the coarsest channel with a valve obstruction, valve-1, with viscosity* $\nu = 1$*.* 36

6.3 *Color map of velocity in the x direction for the coarsest channel with a valve obstruction, valve-1, with viscosity* $\nu = .01$*. Note that negative velocities are present.* 39

6.4 *Color map of velocity in the x direction for the coarsest channel without a valve, channel-1, with* $\mu_{37} = .1$*.* 41

6.5 *Color map displaying viscosity for the coarsest channel without a valve, channel-1, with* $\mu_{37} = .1$*.* 41

6.6 *Color map of velocity in the x direction for the second coarsest channel with a valve obstruction, valve-2, with* $\mu_{37} = .005$*.* 44

6.7 *Color map displaying viscosity for the second coarsest channel with a valve obstruction, valve-2, with* $\mu_{37} = .005$*.* 44

CHAPTER 1
INTRODUCTION

In the field of fluid dynamics, modeling blood flow in a human vein through a venous valve in order to better understand deep vein thrombosis (DVT) is a current area of interest. Thrombosis risk assessment, anti-coagulation therapy, and stroke research would greatly benefit from a better understanding of the microcirculation of blood [8, 11]. When modeling blood flow, several factors are taken under consideration. However, many studies make assumptions about the shear rates of blood flow which can be oversimplifications of the model [9]. The incompressible Navier-Stokes equations are the basis for modeling the flow of an incompressible Newtonian fluid such as water [3] and are given by:

$$\begin{aligned} -\nu\nabla^2\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p &= \mathbf{f}, \\ \nabla\cdot\mathbf{u} &= 0. \end{aligned} \tag{1.1}$$

Although blood can behave as a Newtonian fluid at certain shear rates, it is a non-Newtonian fluid by nature. The Navier-Stokes equations can model a non-Newtonian fluid such as blood when the viscosity term is treated as a function of the scalar strain rate rather than a constant. A number of viscosity models for non-Newtonian blood flow have been developed and this research focuses on the viscosity model developed by Karlene Hoo [11]. Even without non-Newtonian effects, these types of problems are much too complex to compute feasibly by hand. In order to solve a problem modeled by a nonlinear PDE on the computer, first the process of linearization and discretization must take place. As a result of going

through this process, the problem becomes a system of n equations with n unknowns which is represented as the matrix equation Ax=b. For the coupled incompressible Navier-Stokes system, the matrix equation becomes the following block structured system:

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}. \tag{1.2}$$

The solution methods considered here for this system rely on the block structure of the system in order to converge to solutions in such a way that computational time and steps required to solve the system are reduced. Specifically, a physics-based preconditioner called the pressure convection-diffusion preconditioner which utilizes the block structure of the linear system given by (1.2) [2] will be applied to an iterative method used to solve the linear system.

This thesis will discuss the Navier-Stokes equations for Newtonian and non-Newtonian flow, go through the process of linearizing and discretizing the Navier-Stokes equations for both types of flow in order to derive the block structured matrix given by (1.2). Then, approximation techniques for the linear system and the procedure for investigating the effectiveness of the PCD preconditioner on a non-Newtonian flow model will be explained. Finally, results for the PCD preconditioner on Newtonian and non-Newtonian flow will be presented.

CHAPTER 2
NAVIER-STOKES EQUATIONS

The steady-state Navier-Stokes equations are given by:

$$\begin{aligned} -\nu\nabla^2\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p &= \mathbf{f}, \\ \nabla\cdot\mathbf{u} &= 0, \end{aligned} \tag{2.1}$$

with appropriate boundary conditions. As mentioned in the introduction, these equations are the basis for computational modeling of the flow of an incompressible Newtonian fluid [3]. Kinematic viscosity is represented by ν, and is defined as the coefficient of viscosity divided by the density of the fluid [4]. The velocity of the fluid is represented by the vector $\mathbf{u}$ and the pressure is denoted by p. The forcing term $\mathbf{f}$ represents external forces such as gravity acting on the system, and for this research $\mathbf{f} = \mathbf{0}$. From calculus, ∇^2 is the Laplacian operator, ∇ is the gradient operator, and $\nabla\cdot\mathbf{v}$ is the divergence of a vector, $\mathbf{v}$. The convection term is $\mathbf{u}\cdot\nabla\mathbf{u}$, and this is the term that causes the PDE to be nonlinear. The incompressibility constraint is given by the second equation, $\nabla\cdot\mathbf{u} = 0$, which states that the divergence of the fluid velocity is zero.

2.1 Newtonian Flow

Newtonian fluids such as water are characterized by a proportionality relationship between shear stress and strain rate resulting in a constant coefficient of viscosity. Shear stress is defined as a force per unit area. Strain rate, denoted by $\dot{\epsilon}$, is a measure of deformation in terms of relative displacement of the fluid with respect to

time. The coefficient of dynamic viscosity, denoted μ is defined as shear stress divided by strain rate [5]. Since fluid density, denoted by ρ is also a constant, the kinematic viscosity represented by ν, where $\nu = \frac{\mu}{\rho}$ in (2.1) is a constant. With the proper boundary conditions enforced, Newtonian flow through a channel can be modeled by (2.1).

2.2 Non-Newtonian Flow

Non-Newtonian fluids such as blood, ink, and paint do not exhibit a proportionality relationship between shear stress and the strain rate. In other words, a plot of shear stress versus strain rate results in a curve rather than a straight line. Thus, for non-Newtonian fluids, the coefficient of viscosity, μ, is different at each point on the curve and is dependent on the strain rate [5]. As defined above, kinematic viscosity, ν, is given by $\nu = \frac{\mu}{\rho}$, where μ is the coefficient of viscosity and ρ denotes fluid density.

In order to model non-Newtonian flow through a channel, the kinematic viscosity, ν, can no longer be treated as a constant as in Newtonian flow but must be treated as a function of the scalar strain rate, $\dot{\gamma}$, which is determined by the gradient of the fluid velocity, $\nabla\mathbf{u}$. The Navier-Stokes equations for non-Newtonian flow are given by:

$$\begin{aligned} \rho[(\mathbf{u}\cdot\nabla)\mathbf{u}] &= \nabla\cdot[-p\mathbf{I}+\mu(\frac{1}{2}(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))] \\ \nabla\cdot\mathbf{u} &= 0, \end{aligned} \tag{2.2}$$

where $\frac{1}{2}(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)$ is the symmetrized form of the matrix $\nabla\mathbf{u}$ and is multiplied by the function μ. In the case of constant viscosity, it is a beneficial exercise to

demonstrate that the system given above is equivalent to the system given by (2.1) in 1D Cartesian coordinates. Since the second equation in both systems match, equivalence only needs to be shown between the first equation of each system. Consider:

$$\rho[(\mathbf{u}\cdot\nabla)\mathbf{u}] \quad = \quad \nabla\cdot[-p\mathbf{I}+\mu(\frac{1}{2}(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))].$$

Dividing through by ρ, letting $p=\frac{p}{\rho}$, and recalling that $\nu=\frac{\mu}{\rho}$ yields:

$$-\nabla\cdot\nu(\frac{1}{2}(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))+[\mathbf{u}\cdot\nabla\mathbf{u}]+\nabla p=\mathbf{0}.$$

Consider the first term in the simplest case of 1D Cartesian coordinates:

$$\begin{aligned}\nabla\cdot\nu(\frac{1}{2}(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)) &= \frac{1}{2}\frac{\partial}{\partial x}(\nu(\frac{\partial u}{\partial x}+\frac{\partial u}{\partial x}))\\ &= \frac{1}{2}(2\nu\frac{\partial^2 u}{\partial x^2}+2\frac{\partial\nu}{\partial x})\\ &= \nu\frac{\partial^2 u}{\partial x^2}+\frac{\partial\nu}{\partial x}\frac{\partial u}{\partial x}.\end{aligned}$$

This result illustrates that if ν is simply a constant, then $\nabla^2\nu\mathbf{u}=\nu\nabla^2\mathbf{u}$. Further, in this simple example of 1D Cartesian coordinates, for constant ν as in (2.1), $\nabla\cdot\nu(\frac{1}{2}(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))=\nu\nabla^2\mathbf{u}$ showing the desired equivalence between the two forms of the Navier-Stokes equations.

Several models of viscosity for non-Newtonian flow such as the power law model, Walburn-Schneck model, and the Carreau model exist [9]. Models such as these express dynamic blood viscosity as a function of the strain rate, $\dot{\epsilon}$, or more

specifically, the scalar strain rate, $\dot{\gamma}$. Explicity, the strain rate tensor, $\dot{\epsilon}$, is defined as:

$$\dot{\epsilon} = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T),$$

which is the symmetrized velocity gradient tensor. The scalar strain rate, $\dot{\gamma}$, is defined as:

$$\dot{\gamma} = \sqrt{2}\,\|\dot{\epsilon}\|_F = \sqrt{2\dot{\epsilon} : \dot{\epsilon}}$$

where : represents the colon product, also called the dyadic product or Frobenius product, given by $A : B = \sum_{j=1}^{n}\sum_{j=1}^{n} a_{ij}b_{ij}$ for tensors A and B. Thus, the Frobenius norm, $\|\cdot\|_F$, is defined for an n x n matrix A is:

$$\|A\|_F = \left(\sum_{j=1}^{n}\sum_{i=1}^{n} |a_{ij}|^2\right)^{\frac{1}{2}}$$

[10].

The viscosity model for blood based on the principles of minimum energy dissipation was utilized in this research [11]. The model is given by:

$$\mu = \mu_f(1 - \frac{k}{2}C)^{-2}$$

where

$$k = \frac{k_0 + k_\infty(\dot{\gamma}/\dot{\gamma}_c)^{0.5}}{1 + (\dot{\gamma}/\dot{\gamma}_c)^{0.5}}.$$

Dynamic blood viscosity denoted by μ takes into account the viscosity of plasma as well as the concentration of red blood cells. Plasma is a Newtonian fluid, and plasma viscosity is denoted by μ_f which is a constant. The parameters k_0, k_∞, and

$\dot{\gamma}_c$ are functions of the hemotacrit percentage, C, which is the proportion of blood that consists of packed red blood cells. For this research, the hemotacrit percentage is assumed to be 30%. The parameters k_0, k_∞, and $\dot{\gamma}_c$ are expressed by:

$$\begin{aligned} k_0 &= exp(3.874 - 10.41C + 13.8C^2 - 6.738C^3) \\ k_\infty &= exp(103435 - 2.803C + 2.711C^2 - 0.6479C^3) \\ \dot{\gamma}_c &= exp(-6.1508 + 27.923C - 25.6C^2 + 3.697C^3). \end{aligned}$$

These parameters are constants that reflect the suspension properties of blood where k_0 is the intrinsic viscosity of a shear rate near zero, k_∞ is the infinite shear rate or intrinsic viscosity at very high shear rates, and $\dot{\gamma}_c$ is the critical shear rate. For shear rates at or above the critical shear rate, blood behaves as a Newtonian fluid [1]. Under the simplifying assumption that the body is at a constant temperature, the plasma viscosity becomes the constant $\mu_{37} = .0014$ Pa/s , which is the plasma viscosity at $37^\diamond$ Celcius [11].

2.3 Model of Channel and Channel with Valve

For the simplest model, consider the flow of an incompressible fluid between two fixed parallel plates, or a two-dimensional Poiseuille flow model [5]. Poiseuille flow, typically Newtonian, is named after the French physician and physiologist Jean Louis Marie Poiseuille. This model assumes $\mathbf{u} = 0$ along the top and bottom boundary of the channel with a pressure driven flow. At the inlet or left boundary, $p = 1$, and at the outlet or right boundary, $p = 0$. The difference in pressure causes flow to ensue from left to right.

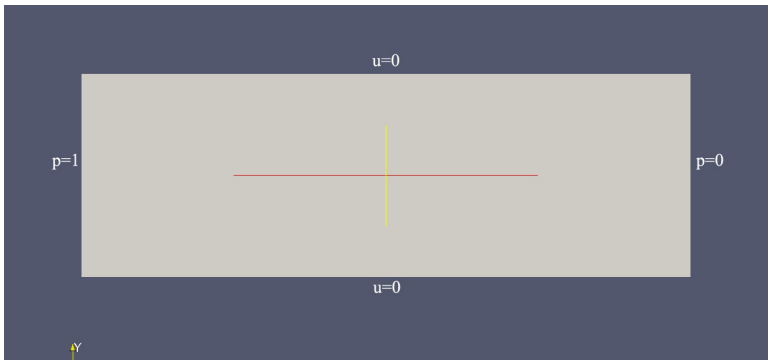

Figure 2.1. *A model of pressure driven flow through a channel with* $\mathbf{u}$*=0 along the top and bottom boundary,* $p = 1$ *at the left boundary, and* $p = 0$ *at the right boundary.*

To move in the direction of modeling the human vein and venous valve, consider the more complicated model including an obstruction (valve) in the channel. The role of a venous valve is to aid in antegrade blood flow towards the heart and prevent retrograde flow due to gravity. The valve is made up of a pair of flexible leaflets which are attached at a point to the vein wall with their free edge protruding into the lumen (interior) of the vein [11]. In reality, both the vein walls and valve are flexible structures. A much simpler model with fixed walls and a fixed valve is

examined in this research at a specific moment in time when the valve is open allowing flow to pass through.

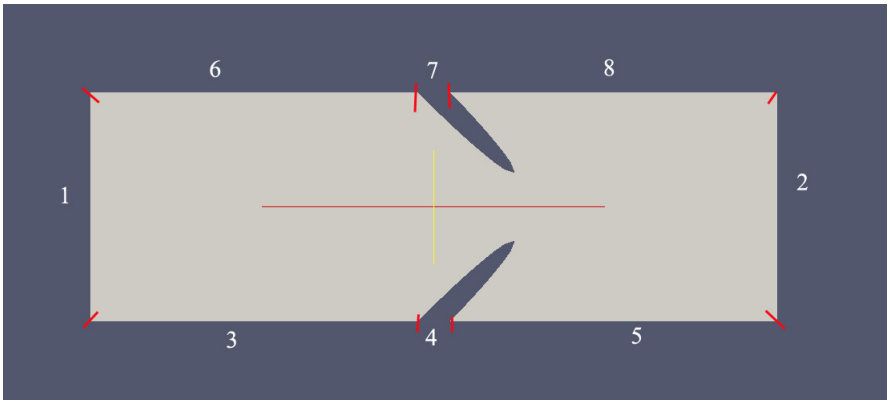

Figure 2.2. *A simple model of pressure driven flow through a channel with fixed top and bottom walls and a fixed obstruction (valve). For regions 3,4,5,6,7,8,* $\mathbf{u} = \mathbf{0}$. *For region 1,* $p = 1$. *For region 2,* $p = 0$.

CHAPTER 3
LINEARIZATION AND DISCRETIZATION

As mentioned in the introduction, the Navier-Stokes equations are a system of continuous nonlinear partial differential equations (PDEs). In order to solve the Navier-Stokes equations on a computer, the system must be transformed into a system of discrete linear equations by the process of linearization and discretization. This must be done for both the Newtonian and non-Newtonian Navier-Stokes equations. Linearization is the process of approximating a nonlinear system with linear equations and discretization involves approximating the infinite-dimensional problem by a finite-dimensional problem. Many options exist for the methods of linearization and discretization and the order is not always set. For this problem, one may linearize and then discretize the system of continuous linear equations as in this research, or one could begin with discretizing the continuous nonlinear PDE and then linearize the discretized nonlinear system.

3.1 Linearization

When it comes to linearization, there are two approaches, which are Newton's method and Picard iteration. While an advantage of Newton's method is quadratic convergence, a disadvantage to Newton's method is that typically the radius of the ball of convergence is proportional to the viscosity parameter, ν. The viscosities of interest for this problem are small, of the order 10^{-2} and 10^{-3}, which makes the task of finding an initial guess within the ball of convergence very difficult. Conversely, the Picard iteration has a "huge ball of convergence," but the rate of convergence is only linear in general [3]. In practice, a good solution technique is to

take a few Picard iterations in order to acquire an initial guess within the ball of convergence for Newton's method. In the non-Newtonian case, there are two nonlinear components: the convection term and the diffusion term since the viscosity is a function of the gradient of velocity. Due to the planned approach to linearize the non-Newtonian viscosity term with a lagging strategy, the natural choice of linearization for this research was to take a Picard iteration.

Consider the Navier-Stokes equations for Newtonian flow, the system given by (2.1). The Picard iteration for the nonlinear system is derived by lagging the coefficient in the quadratic convection term $\mathbf{u} \cdot \nabla \mathbf{u}$. By starting with an initial guess $\mathbf{u^{(0)}}$ for the velocities, the updated velocity and pressure (the k^{th} Picard iterate) can be found by solving the Oseen equations given by:

$$\begin{aligned} -\nu \nabla^2 \mathbf{u}^{(k)} + \mathbf{u}^{(k-1)} \cdot \nabla \mathbf{u}^{(k)} + \nabla p^{(k)} &= \mathbf{f} \\ \nabla \cdot \mathbf{u}^{(k)} &= 0. \end{aligned} \tag{3.1}$$

Since the lagged $\mathbf{u}$, $\mathbf{u}^{(k-1)}$, is known from the previous step, or from the initial guess in the case where $k = 1$, the system is now linear. We stop iterating when the norm of the difference in the current velocity and the previous velocity, $\left\|\mathbf{u}^{(k)} - \mathbf{u}^{(k-1)}\right\|_2$, is less than a chosen tolerance.

Similarly, by starting with an initial guess $\mathbf{u}^{(0)}$ for the velocities in the Navier-Stokes equations for non-Newtonian flow (2.3) , the updated velocity and pressure can be found by solving the following system:

$$\begin{aligned} \rho[(\mathbf{u}^{(k-1)} \cdot \nabla)\mathbf{u}^{(k)}] &= \nabla \cdot [-p^{(k)}\mathbf{I} + \mu(\frac{1}{2}(\nabla \mathbf{u}^{(k)} + (\nabla \mathbf{u}^{(k)})^T))] \\ \nabla \cdot \mathbf{u}^{(k)} &= 0. \end{aligned} \tag{3.2}$$

Unfortunately, (3.2) is not yet linear since μ is a function of the scalar strain rate, $\dot{\gamma}$, which depends on the gradient of the velocity. In the spirit of Picard, lagging μ would complete the linearization of (3.2). Since μ is a function of the scalar strain rate, $\dot{\gamma}$, which depends on the current velocity, $\mathbf{u}^{(k)}$, consider μ_{lag} to be a function of the scalar strain rate, $\dot{\gamma}$, which will now be formed using the previous velocity, $\mathbf{u}^{(k-1)}$. This leads to the following linearized system for non-Newtonian flow:

$$\begin{aligned} \rho[(\mathbf{u}^{(k-1)} \cdot \nabla)\mathbf{u}^{(k)}] &= \nabla \cdot [-p^{(k)}\mathbf{I} + \mu_{lag}(\frac{1}{2}(\nabla \mathbf{u}^{(k)} + (\nabla \mathbf{u}^{(k)})^T))] \\ \nabla \cdot \mathbf{u}^{(k)} &= 0. \end{aligned} \tag{3.3}$$

By discretizing (3.1) for the Newtonian flow and discretizing (3.3) for the non-Newtonian flow, a linear system of equations of the form (1.1) will be produced, which must be solved at each Picard iteration [2].

3.2 Discretization

Each of the systems (3.1) and (3.3) from the previous section are continuous linear systems for Newtonian and non-Newtonian flow, respectively. The point of discretization is to approximate an infinite-dimensional problem by a finite dimensional problem.

The finite difference method and the finite element method are two methods utilized for discretization. Non-uniform meshes as well as complicated domains are very problematic for the finite difference method. For the simple channel model without a valve, the domain is simple and a uniform mesh is easy to obtain, so the finite difference method could be utilized for discretization. As soon as the obstruction (valve) is introduced, it becomes extremely difficult, if not impossible,

to create a uniform mesh. Consider Figure 3.1 illustrating portions of the two coarsest meshes used in this research enlarged around the valve:

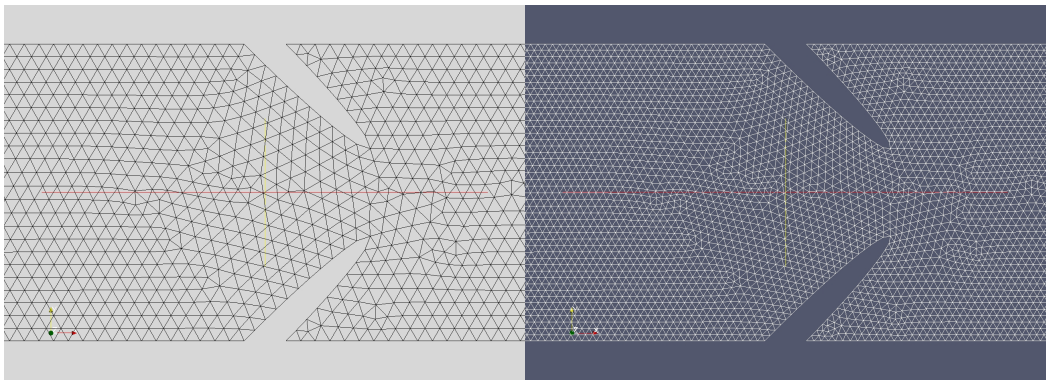

Figure 3.1. *Sample meshes: portion of coarsest mesh is pictured on the left and a portion of the next coarsest mesh is on the right*

These illustrations show that the mesh is not uniform, which is a major reason why the finite element method is used as the method for discretization in this research. The main theorem that is used in this section is the divergence theorem, which says that for a smooth enough vector field $\mathbf{w}$ and any region Ω with a smooth enough boundary $\partial\Omega$, the following result holds:

$$\int_{\Omega} \nabla \cdot \mathbf{w} = \int_{\partial\Omega} \mathbf{w} \cdot \hat{\mathbf{n}},$$

where $d\Omega$ is the increment of area, and $\partial\Omega$ is the increment of the boundary [3]. The definitions of the following spaces will also be needed:

$$L_2(\Omega) := \{\mathbf{u} : \Omega \to \mathbb{R}^2 | \int_{\Omega} \mathbf{u}^2 < \infty\}$$

$$\mathbb{H}^1_{E_0} := \{\mathbf{v} \in \mathcal{H}^1(\Omega)^d | \mathbf{v} = \mathbf{0}\, on\, \partial\Omega_D\}.$$

$L_2(\Omega)$ is the space of functions that are square-integrable in the sense of Lebesgue. The space $\mathcal{H}^1(\Omega)^d$ is a Sobolev space, which is the space of functions in $L_2(\Omega)$ with first derivatives also in $L_2(\Omega)$. The space $\mathbb{H}^1_{E_0}$ is the space of functions in the Sobolev space $\mathcal{H}^1(\Omega)^d$, which are zero on the boundary, $\partial\Omega_D$ [3].

The finite element method will now be applied to discretize each system. The first step is to derive a weak formulation for (3.1) and (3.3). Weak formulations are advantageous in that they lower the order of differentiation on $\mathbf{u}$ making implementation easier. In order to derive the weak form, the following requirements must be satisfied for all $\mathbf{v} \in \mathbb{H}^{\mathbf{1}}_{\mathbf{E_0}}$and $q \in L_2(\Omega)$:

$$\begin{aligned} \int_\Omega \mathbf{v} \cdot (-\nu \nabla^2 \mathbf{u}^{(k)} + \mathbf{u}^{(k-1)} \cdot \nabla \mathbf{u}^{(k)} + \nabla p^{(k)} - \mathbf{f}) &= 0, \\ \int_\Omega q \nabla \cdot \mathbf{u}^{(k)} &= 0. \end{aligned}$$

For ease of notation, let $\mathbf{u_{lag}} = \mathbf{u}^{(k-1)}$, $\mathbf{u} = \mathbf{u}^{(k)}$, and $p = p^{(k)}$. This leads to the equations:

$$\begin{aligned} -\nu \int_\Omega \mathbf{v} \cdot \nabla^2 \mathbf{u} + \int_\Omega \mathbf{v} \cdot (\mathbf{u_{lag}} \cdot \nabla \mathbf{u}) + \int_\Omega \mathbf{v} \cdot \nabla p &= \int_\Omega \mathbf{v} \cdot \mathbf{f}, \\ \int_\Omega q \nabla \cdot \mathbf{u} &= 0. \end{aligned}$$

Now, integrating by parts, applying the divergence theorem, and using the

boundary conditions yields:

$$\begin{aligned} -\nu \int_\Omega \mathbf{v} \cdot \nabla^2 \mathbf{u} &= \nu \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} - \nu \int_\Omega \nabla \cdot (\nabla \mathbf{u} \cdot \mathbf{v}) \\ &= \nu \int \nabla \mathbf{u} : \nabla \mathbf{v} - \nu \int_{\partial\Omega} (\hat{\mathbf{n}} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} \\ &= \nu \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v}. \end{aligned} \tag{3.4}$$

The first step in this result is obtained by rewriting the Laplacian term as $\mathbf{v} \cdot (\nabla \cdot \nabla \mathbf{u})$, and applying the product rule for the divergence of a tensor times a vector. Applying the divergence theorem to $\nu \int_\Omega \nabla \cdot (\nabla \mathbf{u} \cdot \mathbf{v})$ yields $\nu \int_{\partial\Omega} (\hat{\mathbf{n}} \cdot \nabla \mathbf{u}) \cdot \mathbf{v}$. Since $\mathbf{v} \in \mathbb{H}^1_{E_0}$, $\mathbf{v} = \mathbf{0}\, on\, \partial\Omega$, the term from the divergence theorem, $\nu \int_{\partial\Omega} (\hat{\mathbf{n}} \cdot \nabla \mathbf{u}) \cdot \mathbf{v}$ is zero. For the gradient of pressure term, the same process of integrating by parts, applying the divergence theorem, and using the boundary conditions yields:

$$\begin{aligned} \int_\Omega \mathbf{v} \cdot \nabla p &= -\int_\Omega p(\nabla \cdot \mathbf{v}) + \int_\Omega \nabla \cdot (p\mathbf{v}) \\ &= -\int_\Omega p(\nabla \cdot \mathbf{v}) + \int_{\partial\Omega} p\hat{\mathbf{n}} \cdot \mathbf{v} \\ &= -\int_\Omega p(\nabla \cdot \mathbf{v}). \end{aligned}$$

The first result is obtained by using the product rule for the divergence of a scalar times a vector: $\nabla \cdot (p\mathbf{v}) = p(\nabla \cdot \mathbf{v}) + \mathbf{v} \cdot \nabla p$. Applying the divergence theorem to $\int_\Omega \nabla \cdot (p\mathbf{v})$ yields $\int_{\partial\Omega} p\hat{\mathbf{n}} \cdot \mathbf{v}$, and as before this term disappears since $\mathbf{v} \in \mathbb{H}^1_{E_0}$. Therefore, the weak formulation for (3.1) is given by:

$$\begin{aligned} \nu \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} + \int_\Omega (\mathbf{u_{lag}} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} - \int_\Omega p(\nabla \cdot \mathbf{v}) &= \int_\Omega \mathbf{f} \cdot \mathbf{v}, \\ \int_\Omega q \nabla \cdot \mathbf{u} &= 0. \end{aligned}$$

The weak formulation for the Navier-Stokes equations modeling Newtonian flow shown above is a standard result and can be easily found, e.g. see [3]. It is important to note that in the derivation of the weak form above, the viscosity term, ν, was a constant and did not play a significant role. However, when deriving the weak formulation for non-Newtonian flow, the viscosity term is no longer a constant which causes the weak formulation for (3.3) to be more complicated. Let σ represent the following tensor from (3.3), $\sigma = \mu(\frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$. Note that σ is symmetric, so the system given by (3.3) can now be rewritten as follows:

$$\nabla \cdot \sigma + \rho(\mathbf{u_{lag}} \cdot \nabla \mathbf{u}) + \nabla p = \mathbf{0}.$$

The only notable difference in the weak formulation is in the Laplacian term:

$$\begin{aligned} \int_\Omega \mathbf{v} \cdot (\nabla \cdot \sigma) &= \int_\Omega \nabla \cdot (\sigma \mathbf{v}) - \int_\Omega \sigma^T : \nabla \mathbf{v} \\ &= \int_{\partial\Omega} (\sigma \mathbf{v}) \cdot \hat{\mathbf{n}} - \int_\Omega \sigma : \nabla \mathbf{v} \\ &= \int_\Omega \mu(\frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) : \nabla \mathbf{v}. \end{aligned}$$

The first step is obtained by applying the product rule for the divergence of a tensor times a vector given by:

$$\nabla \cdot (A\mathbf{w}) = \mathbf{w} \cdot (\nabla \cdot A) + A^T : (\nabla \mathbf{w}). \tag{3.5}$$

where A is a tensor, and $\mathbf{w}$ is a vector. Applying this rule to the tensor σ and the vector $\mathbf{v}$ yields:

$$\nabla \cdot (\sigma \mathbf{v}) = \mathbf{v} \cdot (\nabla \cdot \sigma) + \sigma^T : \nabla \mathbf{v}.$$

The second step is obtained by applying the divergence theorem, $\int_\Omega \nabla \cdot (\sigma \mathbf{v}) = \int_{\partial\Omega} (\sigma \mathbf{v}) \cdot \hat{\mathbf{n}}$, and the final step once again relies on the definition of the space for the test functions $\mathbf{v}$. In this research, the following approximation was used in the non-Newtonian case:

$$\int_\Omega \mu(\frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) : \nabla \mathbf{v} \approx \int_\Omega \mu \nabla \mathbf{u} : \nabla \mathbf{v}.$$

Therefore, the weak formulation for (3.3) that was used in this research is given by:

$$\begin{aligned} \int_\Omega \mu \nabla \mathbf{u} : \nabla \mathbf{v} + \int_\Omega (\mathbf{u_{lag}} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} - \int_\Omega p(\nabla \cdot \mathbf{v}) &= \mathbf{0}, \\ \int_\Omega q \nabla \cdot \mathbf{u} &= 0. \end{aligned}$$

To apply the Taylor-Hood method of finite elements refers to using biquadratic approximation for the velocity components and bilinear approximation applied to the pressure components. In order to achieve this criteria, second order Lagrange basis functions denoted by $\{\phi_n\}$ are applied to the velocity space while first order Lagrange basis functions denoted by $\{\psi_n\}$ are applied to the pressure space.

Approximating the velocity and pressure components as discrete linear

combinations of the basis functions leads to the following matrices:

$$\begin{aligned} A &= [a_{ij}] = \int_\Omega \nabla\phi_i : \nabla\phi_j \\ N &= [n_{ij}] = \int_\Omega (\mathbf{u_{lag}} \cdot \nabla\phi_\mathbf{j}) \cdot \phi_i \\ B &= [b_{kj}] = -\int_\Omega \psi_k \nabla \cdot \phi_j. \end{aligned}$$

The matrix A is called the vector-Laplacian matrix, the matrix N is the vector-convection matrix, and the matrix B is the divergence matrix. After natural block partitioning, the discrete Oseen problem becomes:

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}.$$

The block $F = \nu A + N$ operates on the discrete velocity space and includes the convection and diffusion terms. The block B^T is the gradient operator and operates on the discrete pressure space, and the block B corresponds to the divergence operator and operates on the discrete velocity space [3].

CHAPTER 4

APPROXIMATING EACH SUBSOLVE

The goal now is to solve the linear system that is produced at each Picard iteration. The system is of the form $A\mathbf{x} = \mathbf{b}$. One approach to solving linear systems is by a direct method such as Gaussian elimination or QR factorization. For a matrix of size $n \times n$, the algorithms for direct methods such as these have a total work estimate of $O(n^3)$ since there are $O(n)$ steps with each step requiring $O(n^2)$ work. One advantage of iterative methods is that for a sparse system they require only $O(n^2)$ operations to solve a linear system instead of $O(n^3)$ operations allowing for much larger matrix problem to be solved on a computer [10]. Another issue that arises is computer memory. In general,matrices produced by the finite element method for PDE's such as in this problem are sparse. Sparse matrices have $O(n)$ entries, and when a direct method such as LU factorization is applied to a sparse matrix, the factors L and U are typically dense, meaning they have $O(n^2)$ entries which must be stored in memory. The same is true for QR factorization. For small matrices, memory is not an issue but as n becomes large, storing $O(n^2)$ entries in memory becomes cumbersome and n eventually becomes large enough that the memory required to store $O(n^2)$ entries exceeds the actual memory capacity of the computer. Iterative methods, on the other hand, only require that $O(n)$ entries be stored in memory for a sparse matrix along with a small number of vectors. Less memory usage is another advantage of iterative methods.

4.1 Iterative Methods

The only requirement for an iterative method algorithm is the capability to determine $A\mathbf{x}$ for any vector $\mathbf{x}$. There are two main categories of iterative methods for solving linear systems — classical methods and Krylov subspace methods. Examples of classical methods are methods such Jacobi and Gauss-Siedel while Krylov subspace methods include methods such as GMRES and Conjugate Gradients. The k^{th} Krylov subspace denoted by $\mathcal{K}_k$ is the space spanned by the vectors $\mathbf{b}$, $A\mathbf{b}$, $A^2\mathbf{b}$, ... , $A^{k-1}\mathbf{b}$:

$$\mathcal{K}_k = \left\langle \mathbf{b},\ A\mathbf{b}, A^2\mathbf{b}, \ldots,\ A^{k-1}\mathbf{b} \right\rangle .$$

When solving the linear system $A\mathbf{x} = \mathbf{b}$, one is looking for the solution $\mathbf{x} \in \mathbb{R}^n$ since A is n x n. The idea behind Krylov subspace methods is to find the "best" approximation to the exact solution $\mathbf{x}$ at each successive subspace $\mathcal{K}_k$.
Schematically:

$$\begin{aligned}
\mathbf{x}_0 &\in \langle \mathbf{b} \rangle \subset \mathbb{R}^n \\
\mathbf{x}_1 &\in \langle \mathbf{b},\ A\mathbf{b} \rangle \subset \mathbb{R}^n \\
&\vdots \\
\mathbf{x}_k &\in \left\langle \mathbf{b},\ A\mathbf{b}, A^2\mathbf{b}, \ldots,\ A^{k-1}\mathbf{b} \right\rangle \subset \mathbb{R}^n
\end{aligned}$$

The criteria by which "best" is measured is one of the main differences between the different Krylov subspace methods [10].

4.1.1 GMRES

The iterative method utilized in this research is the Krylov subspace method GMRES which stands for "generalized minimal residuals." The basic idea of GMRES is as follows: at step k, approximate the exact solution to the problem, $A^{-1}\mathbf{b}$, by the vector $\mathbf{x}_k \in \mathcal{K}_k$ that minimizes the 2-norm of the residual, $r_k = b - A\mathbf{x}_k$. In other words, GMRES is a Krylov subspace method that defines the "best" approximation to the exact solution $\mathbf{x}$ at the k^{th} iteration by finding vector $\mathbf{x}_k \in \mathcal{K}_k$ such that $\|r_k\|_2$ is minimized. Like all Krylov methods, GMRES must typically be preconditioned in order to be successful [10].

4.2 Preconditioning

The basic idea of preconditioning a system of equations is fairly simple. Consider solving the following n x n nonsingular system:

$$A\mathbf{x} = \mathbf{b}, \tag{4.1}$$

such as the linear system that arises from each Picard iteration. Then for any nonsingular matrix M, the system:

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b} \tag{4.2}$$

has the same solution. The advantage is that if the system is solved iteratively, then the convergence will now rely on properties of $M^{-1}A$ instead of properties of A. Thus if the preconditioner M is carefully chosen, the system given in (4.2) may be solved much faster than the system given in (4.1). In this case, M is acting as a left

preconditioner.

Another approach would be to apply a preconditioner on the right by transforming (4.1) into $AM^{-1}\mathbf{y} = \mathbf{b}$ where $\mathbf{x} = M^{-1}\mathbf{y}$. Both left and right preconditioners are used in practice [10]. In this research, a right preconditioner will be applied to each linear system of equations.

4.2.1 ILU Preconditioner

The preconditioner responsible for making the concept of preconditioning famous in the 1970s is the incomplete LU factorization (ILU). When solving the system $A\mathbf{x} = \mathbf{b}$, if the matrix A is sparse, then methods such as LU factorization tend to destroy zeros. For example, if $A = LU$, then the factors L and U will usually be dense. The idea behind the ILU preconditioner is to find an incomplete LU factorization of A by forcing L and U to remain sparse so that $A \approx \tilde{L}\tilde{U}$. Defining a preconditioner $M = \tilde{L}\tilde{U}$ turns out to be a highly effective preconditioner for solving some problems. Several variations of the idea have been proposed and developed extensively [10]. For this problem, ILU fails as a preconditioner. Consider the sparsity pattern of the blocked matrix illustrated by Figure 4.1 and the sparsity patterns of the factors L and U illustrated by Figure 4.2.

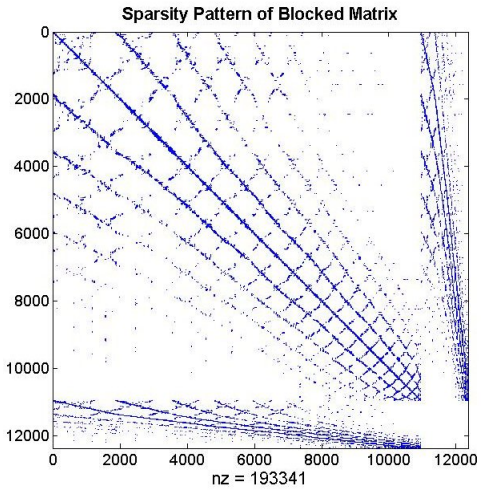

Figure 4.1. *Sparsity pattern of the n x n block matrix from the coarsest valve mesh used in this research. Nonzero entries are shown in blue; all other entries are zero. For this matrix,* $n = 12,373$*, and the matrix has 193,341 nonzeros denoted nz in the figure which is* $O(n)$.

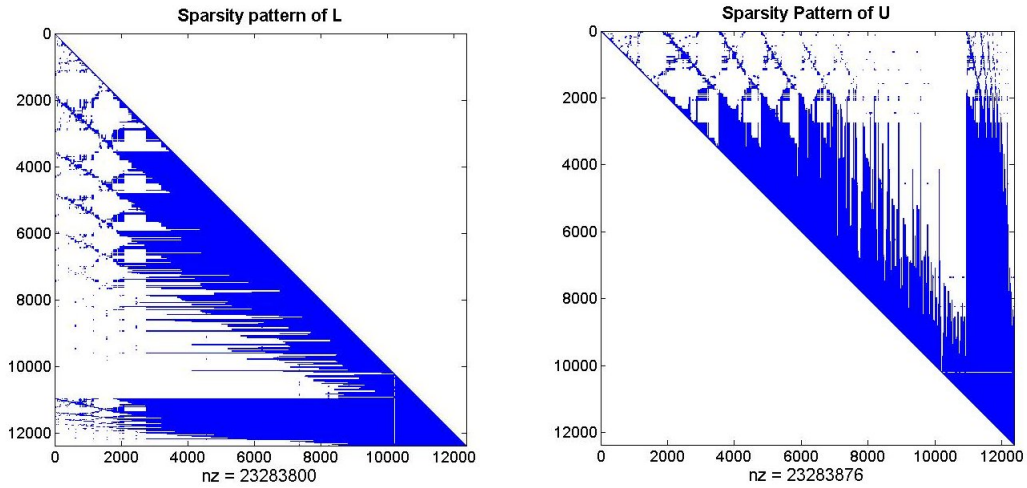

Figure 4.2. *Sparsity pattern of the L and U factors from an LU decomposition of the matrix in figure 4.1. For both L and U, the matrix size is $n = 12,373$, but each factor has approximately 23 million nonzeros, denoted nz in the figure, which is $O(n^2)$ rather than the $O(n)$ number of nonzeros in the original matrix.*

The sparse blocked matrix in Figure 4.1 of size $n = 12,373$ has 193,341 or $O(n)$ nonzero entries, while each of the factors L and U shown in Figure 4.2 have $O(n^2)$ nonzero entries. Specifically, L and U have 23,283,800 and 23,283, 876 nonzero entries, respectively. Since L and U are so dense, it is difficult to make an efficient sparse approximation to L and U. Thus, it is not expected that the ILU preconditioner will perform well on this problem. The density of these L and U factors also indicates that using a direct method such as LU factorization on the systems considered in this thesis will be quite expensive and demonstrates the need for iterative methods.

4.2.2 PCD Preconditioner

The pressure convection-diffusion (PCD) preconditioner is a block preconditioner that was designed to solve linear systems associated with the incompressible Navier-Stokes equations [2]. These linear systems are of the blocked form derived in chapter 2:

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}. \tag{4.3}$$

An LU factorization of the blocked matrix above yields:

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ BF^{-1} & I \end{bmatrix} \begin{bmatrix} F & B^T \\ 0 & -S \end{bmatrix}, \tag{4.4}$$

where $S = BF^{-1}B^T$ is the Schur complement. The Schur complement can be derived by performing one step of Gaussian elimination on the blocked 2 x 2 system and observing the entry in the bottom right block of the matrix. Now, (4.4) implies that:

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} F & B^T \\ 0 & -S \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ BF^{-1} & I \end{bmatrix}. \tag{4.5}$$

Since the resulting matrix in (4.5) is lower triangular with all eigenvalues along the diagonal with value 1, this suggests that the matrix

$$Q = \begin{bmatrix} F & B^T \\ 0 & -S \end{bmatrix}$$

is an effective right preconditioner for the blocked linear system given by (4.3) [2]. For a preconditioner such as Q to be used in a Krylov subspace iteration, each step

requires the application of Q^{-1} to a vector. To illustrate the computational issues involved in applying the preconditioner Q at a specific step (particular Krylov iteration), consider Q^{-1} in factored form:

$$Q^{-1} = \begin{bmatrix} F & B^T \\ 0 & -S \end{bmatrix}^{-1} = \begin{bmatrix} F^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & -B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & -S^{-1} \end{bmatrix}.$$

Thus, applying Q^{-1}as a preconditioner will require the application of S^{-1} to a vector in the discrete pressure space as well as the application of F^{-1} to a vector in the discrete velocity space. Since these operations are too expensive for a practical computation, an effective preconditioner can be derived by replacing these two operations with inexpensive approximations [2]. Although only valid in special cases, supposing that there is an analogous convection-diffusion operator defined on the discrete pressure space, $F_p = -\nu\nabla^2 p + \mathbf{u_{lag}} \cdot \nabla p$, and making the assumption that the convection-diffusion operators formally commute with the gradient operator suggests a convenient approximation for S. These assumptions yield the following:

$$FB^T \approx BF_p$$

Now by algebraic manipulation:

$$\begin{aligned} B^T &\approx F^{-1}BF_p \\ B^T F_p^{-1} &\approx F^{-1}B \\ BB^T F_p^{-1} &\approx S. \end{aligned}$$

So an approximation to the Schur complement, S, is $\hat{S} \approx (BB^T)F_p^{-1}$ which implies $S^{-1} \approx F_p(BB^T)^{-1}$ [2]. In this research, the PCD preconditioner is applied in the solution of the linear systems that arise in each step of the nonlinear (Picard) iteration for modeling Newtonian and non-Newtonian flow through a channel and channel with valve.

CHAPTER 5
APPLICATION OF PRECONDITIONERS TO FLOW THROUGH A CHANNEL PROBLEM

In this chapter, the focus is to describe the methods implemented in order to investigate the effectiveness of the PCD preconditioner on both the Newtonian and non-Newtonian models of flow through a channel as well as flow through a channel with a valve. The methods for investigation were executed in C++ using Trilinos. The chapter will begin by describing the software packages used in this research and then transition to the set up of the computational experiments.

5.1 Trilinos and Sundance

A project of Sandia National Laboratories, the Trilinos Project focuses on developing and implementing robust algorithms by enabling technologies using modern object-oriented software design while still taking into account the value of established libraries [6]. Trilinos, meaning "a string of pearls," is essentially a collection of software packages that have the ability to utilize one another and communicate with multiple libraries. Each package has a specific purpose such as preconditioning, solution of linear systems, solution of eigenvaule problems, discretization, etc. The main package utilized for this project is Sundance [7]. Sundance is a system for rapid development of high-performance finite-element solutions of partial differential equations. A major advantage and convenience of Sundance is that is was created with the idea that one should be able to code a finite element problem on a computer with the same level of abstraction that would be used in describing the problem on a blackboard. Other packages used were

Amesos, which is a package of direct linear solvers such as LU factorization, and Belos, which is a package that provides iterative methods such as GMRES.

5.2 Procedure

The procedure to investigate the effectiveness of the PCD preconditioner consists of three main steps, each associated with the development of a specific Sundance code. As a first step, incompressible Newtonian flow is modeled with the Navier-Stokes equations. The Navier-Stokes equations are linearized to produce the Oseen equations, and LU factorization is used to solve the linear system produced at each Oseen step. For Newtonian flow, viscosity is a constant, and this value is set in the code. Second, Newtonian flow is again modeled with the Navier-Stokes equations which are linearized to produce the Oseen equations. However, for this approach, the linear system produced at each Oseen step is solved by the iterative method GMRES. GMRES is preconditioned with the PCD preconditioner, and the viscosity is a constant once again since Newtonian flow is being modeled. For the third and final step, an incompressible non-Newtonian flow is modeled with the Navier-Stokes equations. After obtaining the Oseen equations by linearization, velocity and viscosity are updated at each Oseen step. GMRES preconditioned with the PCD preconditioner is applied to solve the linear system at each Oseen step.

5.2.1 Step 1: Newtonian flow with direct linear solver

As a first step, code was produced in Sundance to model incompressible Newtonian flow through a channel and a channel with valve. An Oseen loop was put into the code and set to stop when $\|\mathbf{u} - \mathbf{u}_{lag}\| < 10^{-6}$. The linear system produced at each iteration of the Oseen loop was solved by LU factorization, a

direct method in the Amesos package. The weak form for Newtonian flow derived in chapter 3 was given by:

$$\begin{aligned} \nu \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} + \int_\Omega (\mathbf{u_{lag}} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} - \int_\Omega p(\nabla \cdot \mathbf{v}) &= \int_\Omega \mathbf{f} \cdot \mathbf{v}, \\ \int_\Omega q \nabla \cdot \mathbf{u} &= 0. \end{aligned}$$

Since $\mathbf{f} = \mathbf{0}$ and the test functions are linearly independent, the weak form can be writ ten as:

$$\int_\Omega \nu \nabla \mathbf{u} : \nabla \mathbf{v} + (\mathbf{u_{lag}} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} - p(\nabla \cdot \mathbf{v}) + q \nabla \cdot \mathbf{u} = 0 \tag{5.1}$$

This corresponds to the following portion of Sundance code:

```
Expr eqn = Integral(interior, nu * colonProduct(outerProduct(grad, v),
outerProduct(grad, u)) + vx*(lagU*grad)*ux + vy*(lagU*grad)*uy - p*div(v)
- q*div(u)
```

The similarity between this portion of Sundance code and the mathematical statement of the integral in (5.1) illustrates one of the main benefits of using Sundance.

5.2.2 Step 2: Newtonian flow with preconditioned iterative solver

Once the Newtonian code had been developed and was validated on a problem with a known solution, the next logical step was to change the method of solving the linear system produced at each Oseen iteration from a direct method to an iterative method. One change consisted of swapping the Amesos package for the

Belos package, which is a package of iterative methods for solving linear systems. In particular, GMRES was the iterative method used from the Belos package. The matrix formed by Sundance at each step of the Oseen loop does not automatically have the desired blocked structure which the PCD preconditioner relies on to be effective. Grouping the velocity and pressure variables appropriately in the definition of the PDE equation in Sundance produces a linear system with the equations ordered so that the solution vector, $\mathbf{x}$, is ordered as follows:

$$\begin{bmatrix} u_{x_1} \\ u_{y_1} \\ p_1 \\ u_{x_2} \\ u_{y_2} \\ p_2 \\ \vdots \\ \vdots \end{bmatrix} \Longrightarrow \begin{bmatrix} u_{x_1} \\ u_{y_1} \\ u_{x_2} \\ u_{y_2} \\ \vdots \\ p_1 \\ p_2 \\ \vdots \end{bmatrix}$$

After these updates, this code modeled Newtonian flow with the incompressible Navier-Stokes equations which are linearized to produce the Oseen equations. An iterative solver, GMRES preconditioned with the PCD preconditioner, was used to solve the blocked linear system produced at each Oseen step. The maximum GMRES iterations for each Oseen step were set at 500, and the iterations were set to stop when the 2-norm relative residual became less than 10^{-7}. In other words, the stopping criteria is based on the 2-norm of the current residual divided by the 2-norm of the initial residual, $\frac{\|r\|_2}{\|r_0\|_2}$, where $r_0 = \mathbf{b}$.

Unpreconditioned GMRES is not expected to do well on this problem. But in order to confirm, an unpreconditioned GMRES solve was attempted on the easiest

problem considered: the coarsest mesh of the channel with no obstruction with $\nu = 1$. It did not converge within 500 iterations on the first Oseen step which is simply solving the Stokes equations. In addition, it was not close to converging after the 500 iterations of GMRES, and appeared that it would not be close to converging until 1500 or more iterations were taken. As expected, unpreconditioned GMRES did not perform well. Since these results showed little promise, no further runs with unpreconditioned GMRES were attempted.

The viscosity for Newtonian flow is constant, so this value was set (hard-coded). Various viscosities were tested: $\nu = 1,\ 0.1,\ 0.01,\ .001$ — the actual viscosity of blood is of the order 10^{-3}. The first and second code were tested on four different mesh sizes of the channel and the channel with valve, and Table 5.1 reports the matrix sizes corresponding to each mesh:

Table 5.1. *Size, n, corresponding to each of the meshes is reported. Channel refers to the simple channel model (see Figure 2.1) while valve refers to the model of the channel with an obstruction (see Figure 2.2).*

	Size, n		Size, n
channel-1	12,850	valve-1	12,373
channel-2	50,591	valve-2	48,503
channel-3	200,755	valve-3	170,530
channel-4	799,811	valve-4	764,243

Note that the size of the matrices corresponding to mesh 4, with or without the valve obstruction, are near 800,000. This is a very large problem for a single

processor machine, and in the course of this research no results were obtained for these problem sizes due to memory issues.

5.2.3 Step 3: Non-Newtonian flow with preconditioned iterative solver

The final step was to make the transition from Newtonian flow to non-Newtonian flow, which required changing the formulation of the viscosity term in the PDE and the boundary conditions. The viscosity model was an important addition to the code. As before, an Oseen loop was used but for this phase also included lagging the viscosity term as discussed in the linearization section in chapter 3. The Picard iterations were set to stop when $\|\mathbf{u} - \mathbf{u}_{lag}\|_2 < 10^{-6}$, and the maximum number of Picard iterations was set at 20. The iterative solver GMRES preconditioned with the PCD preconditioner was applied to solve the linear system produced at each step of the Picard iteration. The maximum GMRES iterations for each Oseen step were set at 500, and the iterations were set to stop when the relative residual became less than 10^{-7} as in step 2. This code was tested on the two coarsest mesh sizes for the channel and the two coarsest mesh sizes for the channel with valve at various values of μ_{37}.

CHAPTER 6
RESULTS FROM INVESTIGATION

Table 6.1 reports the results for the first step of the procedure. This step modeled Newtonian flow using the Navier-Stokes equations, Oseen linearization, and a direct solver on the linear subproblems. In particular, LU factorization was used as the direct solver for this step of the procedure. Shown in Table 6.1 is the number of nonlinear (Picard) iteration steps taken for $\|\mathbf{u} - \mathbf{u}_{lag}\|_2 < 10^{-6}$ to be satisfied.

Table 6.1. *Results from Newtonian flow model with Oseen linearization and LU factorization applied to linear subproblems at various viscosities for different mesh sizes.*

Viscosity	1	.1	.01	.001
	# of Picard its	# of Picard its	# of Picard its	# of Picard its
channel-1	2	2	2	N
channel-2	2	2	2	N
channel-3	2	2	2	N
valve-1	3	5	N	N
valve-2	3	5	N	N
valve-3	3	5	N	N

The failure mode, N, represents the case where 20 Picard iterations were taken and the convergence criteria for the Oseen loop was not met. For the runs that failed in this manner, the Oseen iterations were not making significant progress in decreasing $\|\mathbf{u} - \mathbf{u}_{lag}\|_2$, and the iterations seemed to stagnate. The physical problem is increasing in difficulty as viscosity decreases, and the problem size is growing as

finer meshes are used within each model. For the channel without a valve, there was no growth in the number of Picard iterations either with increasing problem size or increasing difficulty (decreasing viscosity), but at a viscosity of 0.001, the nonlinear solver no longer converged. The channel with a valve obstruction was more difficult for the nonlinear iteration, and the number of Picard iterations was greater than it was for the unobstructed channel. The number of Picard iterations also increased with decreasing viscosity, and the nonlinear solver began to fail sooner with decreasing viscosity than in the unobstructed case.

The typical channel result displaying velocity in the x direction is illustrated by figure 6.1.

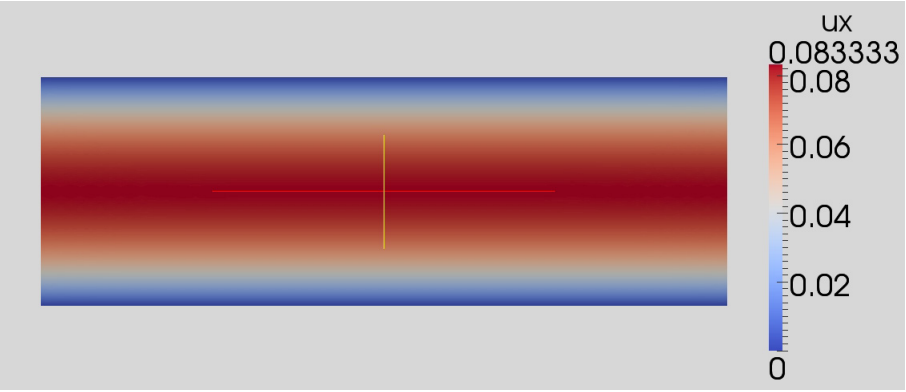

Figure 6.1. *Color map of velocity in the x direction for the coarsest channel without a valve, channel-1, with viscosity* $\nu = 1$.

The color map is displaying a parabolic profile for velocity. Parabolic velocity profiles are a characteristic of Poiseuille flow.

The typical valve result displaying velocity in the x direction is illustrated by figure 6.2.

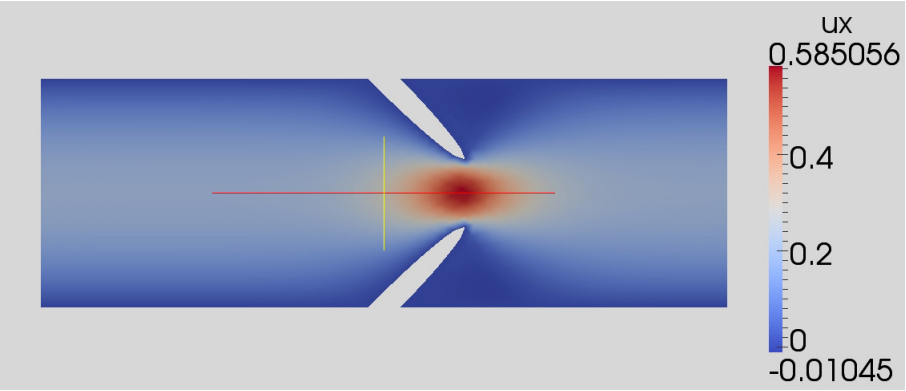

Figure 6.2. *Color map of velocity in the x direction for the coarsest channel with a valve obstruction, valve-1, with viscosity* $\nu = 1$.

Note that the velocities are higher in the narrow part of the channel (going through the valve) as expected since the conservation of mass is preserved by the incompressibility constraint in the Navier-Stokes equations. The same amount of fluid must flow through this narrow part of the channel as flows through the wider regions, thus the velocities must increase here since the fluid is incompressible.

Results for the second step of the procedure are reported by Table 6.2. This step modeled Newtonian flow using the Navier-Stokes equations with Oseen linearization and used GMRES preconditioned with the PCD preconditioner to solve the linear system produced at each Oseen step. The table shows the number of Picard iterations taken for the 2-norm of change in velocity to become less than 10^{-6}. For problems where the nonlinear solve converged, the table also shows the average number of GMRES iterations per Oseen step. The GMRES solves have a stopping tolerance of 10^{-7}.

Table 6.2. *Results from Newtonian flow model with preconditioned GMRES at various viscosities for different mesh sizes. Avg GMRES is the average of the number of GMRES iterations at each Oseen step. N indicates that GMRES did not converge within 500 iterations.*

Viscosity	1		.1		.01	
# of its	Picard	Avg GMRES	Picard	Avg GMRES	Picard	Avg GMRES
channel-1	2	16	2	42	N	
channel-2	2	16	2	41	N	
channel-3	2	16	2	41	N	
valve-1	3	24	5	33	20	240
valve-2	3	24	5	34	N	
valve-3	3	24	5	35	N	

The failure mode, N, represents the case where 500 iterations of GMRES were taken and the convergence criteria was not met. For the cases where N was reported, GMRES stagnated so increasing the maximum iterations of GMRES would not have

helped. The number of average GMRES iterations does not include GMRES iterations needed for the first Oseen step. For the Oseen step, the initial guess $\mathbf{u} = 0$ reduces the problem to the Stokes equations which is an easier problem, and GMRES typically took fewer iterations for this step. As before, the physical problem is increasing in difficulty as viscosity decreases. In the case of the flow in a channel with a valve obstruction, more nonlinear iterations are needed as viscosity decreases. An even smaller viscosity of .001 was also tried but GMRES did not converge for any of the cases. The problem size is growing as finer meshes are used within each model. Scalability, or mesh independence, of the PCD preconditioner is observed in Table 6.2 since the number of GMRES iterations is not growing with problem size. As discussed in the procedure, unpreconditioned GMRES was also tested at this phase in the research. For channel-1 with $\nu = 1$, GMRES did not converge within 500 iterations on the first (0th) Picard iteration. There was no reason to believe that unpreconditioned GMRES would perform well in the first place, and it was not tested for any other cases.

Figure 6.3 illustrates an interesting result obtained from the second phase of the research. Negative velocities are present possibly indicative of blood pooling behind the valve (or on top of the valve in the vertical case).

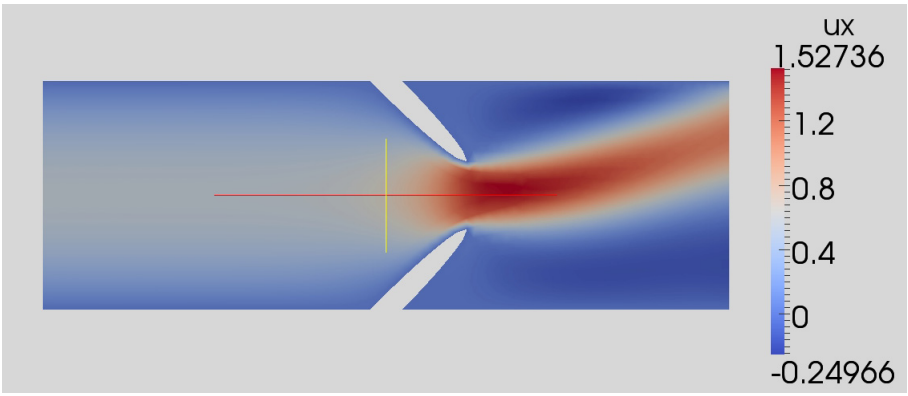

Figure 6.3. *Color map of velocity in the x direction for the coarsest channel with a valve obstruction, valve-1, with viscosity* $\nu = .01$. *Note that negative velocities are present.*

For the third step of the procedure with non-Newtonian flow, an Oseen loop, and GMRES preconditioned with the PCD preconditioner on the channel model, the results are reported by Table 6.3. Results were obtained for channel-1 and channel-2 with $\mu_{37} = .1$ and $\mu_{37} = .01$. Table 6.3 shows the number of nonlinear (Picard) iterations needed in order for $\|\mathbf{u} - \mathbf{u}_{lag}\|_2 < 10^{-6}$ to be satisfied. In addition, it shows the average number of GMRES iterations per nonlinear step, and as before, the first nonlinear iteration was not included in the average. In Table 6.3 as well as Table 6.4, the 2-norm of the initial viscosity field, $\|\mu_0\|$, is recorded as a measure of the magnitude of the viscosity values corresponding to each μ_{37}.

Table 6.3. *Results from the non-Newtonian flow model on the channel with preconditioned GMRES at various viscosities for different mesh sizes. Avg GMRES is the average of the number of GMRES iterations at each Oseen step.*

μ_{37}, $\|\mu_0\|$	.1, 5.096016e+01		.01, 5.096016e+00	
	Picard	Avg GMRES	Picard	Avg GMRES
channel-1	14	16	16	78
channel-2	14	17	16	85

The plasma viscosity, μ_{37}, was the parameter that was changed for the non-Newtonian model. In reality, $\mu_{37} = .0014$ [Hoo]. The values for μ_{37} used in this research were chosen with the intent of approaching the actual value of .0014. The physical problem increased in difficulty as μ_{37} decreased, and the problem size increased as finer meshes were used. Transitioning from the Newtonian model to the non-Newtonian model, both the nonlinear (Picard) and linear (GMRES) iteration counts grew significantly.

A typical channel result for non-Newtonian flow displaying velocity in the x direction is given by Figure 6.4.

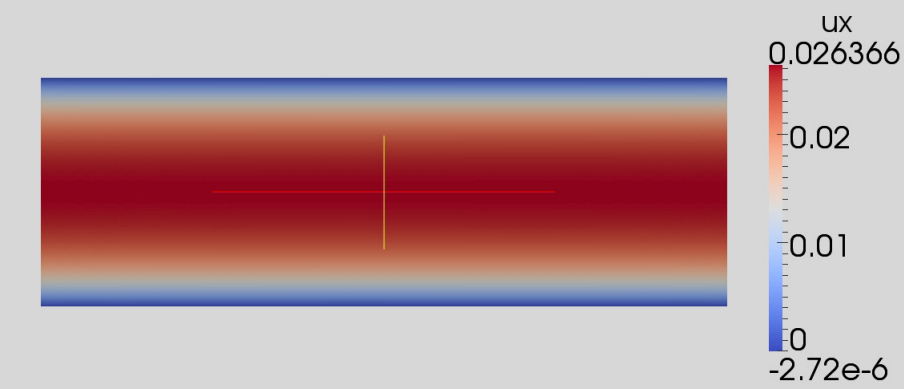

Figure 6.4. *Color map of velocity in the x direction for the coarsest channel without a valve, channel-1, with* $\mu_{37} = .1$.

The color map in Figure 6.4 shows a parabolic profile of velocities for the simple channel result just as the channel result for Newtonian flow illustrated. Figure 6.5 shows a color map of the viscosity for the same problem.

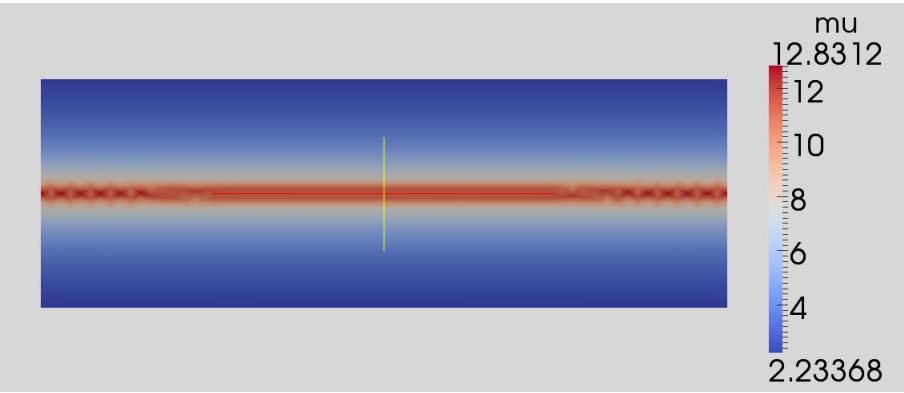

Figure 6.5. *Color map displaying viscosity for the coarsest channel without a valve, channel-1, with* $\mu_{37} = .1$.

Note that the viscosity is not a constant which means that the flow modeled is indeed non-Newtonian. Also, the velocity is highest where the velocity gradient is smallest.

For the third step of the procedure with non-Newtonian flow, an Oseen loop, and GMRES preconditioned with the PCD preconditioner on the channel with valve model, the results are reported by Table 6.4. Results were obtained for valve-1 and valve-2 with $\mu_{37} = .1$, $\mu_{37} = .01$, and $\mu_{37} = .005$. Table 6.4 shows the number of nonlinear (Picard) iterations needed in order for $\|\mathbf{u} - \mathbf{u}_{lag}\|_2 < 10^{-6}$ to be satisfied. In addition, it shows the average number of GMRES iterations per nonlinear step, and as before, the first nonlinear iteration was not included in the average.

Table 6.4. *Results from the non-Newtonian flow model on the valve with preconditioned GMRES at various viscosities for different mesh sizes. Avg GMRES is the average of the number of GMRES iterations at each Oseen step.*

μ_{37}, $\|\mu_0\|$	.1, 5.029571e+01		.01, 5.029571e+00		.005, 2.514786e+00	
	Picard	Avg GMRES	Picard	Avg GMRES	Picard	Avg GMRES
valve-1	11	22	15	83	20	160
valve-2	11	22	16	88	20	195

Transitioning from the Newtonian model to the non-Newtonian model, both the nonlinear (Picard) and linear (GMRES) iteration counts grew significantly.

Again, the physical problem became more difficult as μ_{37} decreased, and the problem size grew as finer meshes were used. Scalability of the PCD preconditioner is observed although moderate growth is observed in the number of linear (GMRES) iterations as the problem size increased for $\mu_{37} = .01$ and $\mu_{37} = .005$. Figure 6.6 shows a typical valve result for non-Newtonian flow displaying velocity in the x direction.

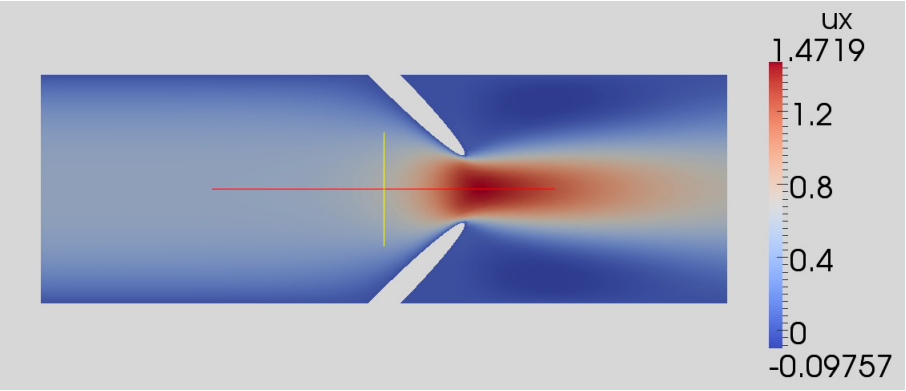

Figure 6.6. *Color map of velocity in the x direction for the second coarsest channel with a valve obstruction, valve-2, with* $\mu_{37} = .005$.

Figure 6.7 shows a color map of the viscosity for the same problem. Once again, note that the viscosity is not a constant which means that the flow modeled is non-Newtonian.

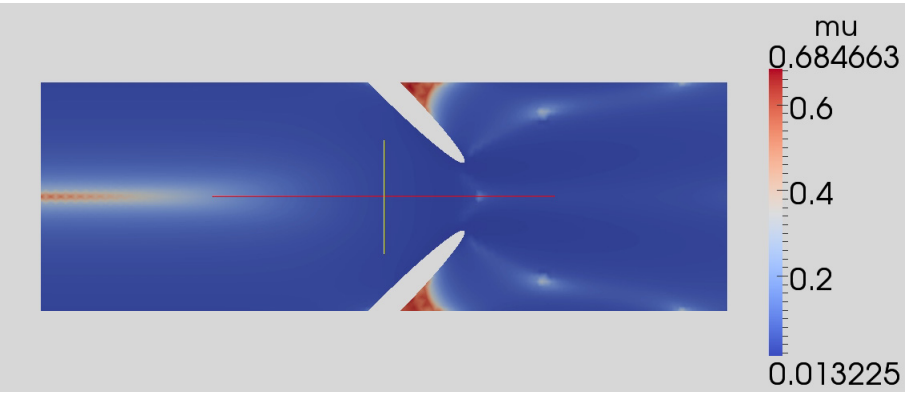

Figure 6.7. *Color map displaying viscosity for the second coarsest channel with a valve obstruction, valve-2, with* $\mu_{37} = .005$.

Note that the viscosity is higher in the corners behind the valves where the gradient of velocity is low. This is consistent with the deep vein thrombosis problem

currently being studied. Where the viscosity is higher, blood may be more likely to clot.

CHAPTER 7
CONCLUDING REMARKS

The primary aim of this research was to apply the PCD preconditioner to a non-Newtonian flow model in order to investigate effectiveness. This involved implementation of Newtonian and non-Newtonian models in C++ through Sundance. The pressure convection-diffusion preconditioner was effective on the non-Newtonian model. The PCD preconditioner showed scalability when applied to both the Newtonian and non-Newtonian flow models. Growth in the linear iteration count (GMRES iterations) occurred as the viscosity decreased for both the Newtonian and non-Newtonian cases. When transitioning from the Newtonian to non-Newtonian model, severe growth was observed in the nonlinear iterations (Picard iterations). This conclusion suggests that Newton's method may be needed for the non-Newtonian model to successfully model more realistic (more difficult physically) blood flow.

There are several natural directions that this work could take in the future. One is developing a model with more realistic geometry for the vein model. Also, implementing a transient problem to observe how flow evolves would be beneficial. Another direction which was mentioned in chapter 3 would be a different approach to the problem. This would involve taking a few Picard iterations in order to get a "good" initial guess for Newton's method in order to utilize the quadratic convergence rate of Newton's method. Although not done in this research, yet another direction would be to run the Sundance codes for these models in parallel. Finally, applying different preconditioners that have been developed for the

incompressible Navier-Stokes equations modeling Newtonian flow to the non-Newtonian model would be another interesting direction.

BIBLIOGRAPHY

[1] G. Cokelet, *Handbook of Bioengineering*. McGraw-Hill, New York, 1987, ch. The Rheology and Tube Flow of Blood.

[2] H. C. Elman, V. E. Howle, J. B. Shadid, and R. S. Tuminaro, *A parallel block multi-level preconditioner for the 3D incompressible Navier-Stokes equations*, Journal of Computational Physics, 187 (2003), pp. 504-523.

[3] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: with applications in incompressible fluid dynamics*, Oxford University Press, New York, 2005.

[4] Y. C. Fung, *Biodynamics: Circulation*, New York, NY: Springer-Verlag, 1984.

[5] E. Guyon, J. Hulin, L. Petit, and C. D. Mitescu, *Physical Hydrodynamics*, Oxford University Press, New York, 2001.

[6] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley, *An Overview of the Trilinos Project*, ACM Transactions on Mathematical Software, 31 (2005), pp. 397-493.

[7] K. Long, R. C. Kirby, and B. van Bloemen Waanders, *Unified Embedded Parallel Finite Element Computations via Software-Based Frechet Differentiation*, SIAM Journal of Scientific Computing, 32 (2010), p. 3323.

[8] A. Rahimian, I. Lashuk, S. K. Veerapaneni, A. Chandramowlishwaran, D. Malhotra, L. Moon, R. Sampath, A. Shringarpure, J. Vetter, R. Vuduc, D. Zorin, and G. Biros, *Petascale direct numerical simulation of blood flow on 200K cores and heterogeneous architectures*, ACM/IEEE SCXY Conference.

[9] M. W. Siebert and P. S. Fodor, *Newtonian and Non-Newtonian Blood Flow over a Backward Facing Step - A Case Study*, Excerpt from the Proceedings of the COMSOL Conf., 2009, Boston, MA.

[10] L.N. Trefethen, and D. Bau III, *Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, 1997.

[11] N. Wijeratne and K. Hoo, *Numerical Studies on the hemodynamics in the human vein and venous valve*, in Proc. American Automatic Control Conf., June 2008, Seattle, WA, paper WeA0-4.