Pennsylvania State University
The Graduate School
College of Engineering

COUPLED FLOW-BIOCHEMISTRY SIMULATIONS OF DYNAMIC SYSTEMS OF BLOOD CELLS

A Thesis in
Mechanical Engineering
by
Byron J. Gaskin

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2014

The thesis of Byron J. Gaskin was reviewed and approved* by the following:

Brent Craven
Professor of Mechanical Engineering
Thesis Co-Advisor

Robert F. Kunz
Senior Scientist, Applied Research Lab
Professor of Aerospace Engineering
Thesis Co-Advisor

Laura Pauley
Professor of Mechanical Engineering

Daniel Haworth
Professor-In-Charge of MNE Graduate Programs
Professor of Mechanical Engineering

Karen Thole
Department Head of Mechanical and Nuclear Engineering
Professor of Mechanical Engineering

*Signatures are on file in the Graduate School.

Abstract

With the aim of improved modeling of biological cell flow simulations, a computational tool has been developed to model a heterogeneous system comprised of an arbitrary number of bodies of arbitrary geometry interacting in highly viscous flows. Accordingly, this model couples computational fluid dynamics (CFD), six degree-of-freedom (6DOF) motion, and surface biochemistry. Adaptive mesh refinement is used to ensure adequate resolution of flow features when multiple surfaces are in close proximity.

The particular contributions of this thesis include: 1) stability analysis of the coupled 6DOF-hydrodynamics system, leading to an efficient adaptive timestep specification, 2) introduction of the immersed boundary method (IBM), wherein arbitrary cell shapes are meshed internally, to enable exact inertial property evaluations, and 3) generalization to an n-body system including a novel cyclic method for reintroducing cells on periodic boundaries to enable statistical stationarity.

These and other improvements presented in this thesis have led to a tool with improved modeling accuracy and computer run times that are orders of magnitude faster than its predecessor.

The capabilities of this tool are demonstrated by presenting several three-dimensional cell system simulations. The presented cell system simulations are found to be in agreement with data found in the literature.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Symbols

2D	Two dimensional
3D	Three dimensional
6DOF	Six degree-of-freedom
FSI	Fluid-Structure Interaction
Re	Reynolds Number
C_{Drag}	Coefficient of Drag
t	Time
t_0	Initial Time
Δt	Timestep
T	Temperature
L	Reference Length
U	Reference Velocity
D	Diameter
r	Radius
ρ	Density
μ	Viscosity
P	Pressure , Probability

m	Mass
A	Area
V	Volume
I	Principal moment of Inertia
u_i	i-th component of velocity vector
x_i	i-th spatial coordinate
θ_i	i-th component of angular displacement vector
r_i	i-th component of radius vector
g_i	Gravity along i-th spatial coordinate
$\hat{n_i}$	Unit normal vector of i-th face
$\hat{e_i}$	Unit vector along line of action of i-th bond
δ_{ij}	Kronecker delta
τ_{ij}	Viscous shear along the i-th face acting in the direction of the j-th spatial coordinate
F_i	Force along i-th spatial coordinate
T_i	Torque applied to the centroid about i-th spatial coordinate
C_i	i-th component of centroid
$\dot{\gamma}$	Flow shear rate
b	Nonlinear spring constant
d	Distance between two faces
k_{on}	Affinity of a molecule to form a bond
k_{on}^0	Affinity of a molecule to form a bond under equilibrium conditions
k_{off}	Affinity of a molecule to break a bond
k_{off}^0	Affinity of a molecule to break a bond under equilibrium conditions

A_L	Contact area
n_L	Number of molecules within a surface area
n_B	Number of bound molecules within a surface area
s	Spring constant of bound molecules
s_{ts}	Spring constant of unbound molecules
λ	Critical distance
k_b	Boltzmann constant
ϵ	Critical distance

Acknowledgments

I would first like to thank my advisor, Dr. Rob Kunz. His guidance has helped me grow personally and professionally, and increased my knowledge as an engineer. I would also like to thank Dr. Cheng Dong for his valuable insight throughout this project and Dr. Brent Craven for helping me navigate through the Mechanical Engineering department.

I would like to thank all of my friends and co-workers at the Applied Research Lab (ARL) for their knowledge, their support, and fun times. I would like to thank my partner-in-crime throughout this project Julie Behr, who, at the time of this writing, is in a better place...New York City...

I would like to express my gratitude to the ARL OpenDOOR program for introducing me to Penn State ARL and to the ARL Walker Graduate Assistantship for providing financial support.

Lastly, I would like to thank my family for their continued support and love. They help me overcome every obstacle and push me to accomplish all of my goals, no matter how ridiculous.

Chapter 1

Introduction

Understanding the social behavior of biological cells in inhomogeneous flow systems is of great importance in the diagnosis and treatment of various medical conditions. As in many physical systems, social interaction can have observable implications on system behavior and dynamics. This phenomenon is prevalent in a variety of natural systems so it may come as no surprise that the effects of socialization in cellular systems are equally noticeable.

In addition to intra-cellular social interactions, flow environments of a cell can have direct impacts on its adhesive and geometric properties; in turn, these impacts may affect its functionality. One example in which cellular geometry compromises cellular functionality is sickle cell anemia [1, 2, 3, 4]. Sickle cell anemia is attributed to an abnormal type of hemoglobin inside red blood cells. This abnormal protein changes the geometrical shape of the red blood cell, causing the cells to become crescent or sickle-shaped. Due to this geometric abnormality, the cells are much more fragile and deliver lower amounts of oxygen to body tissue. In addition to its functional deficiencies, these sickle-shaped cells are prone to blocking small blood vessels and may burst, interrupting normal blood flow.

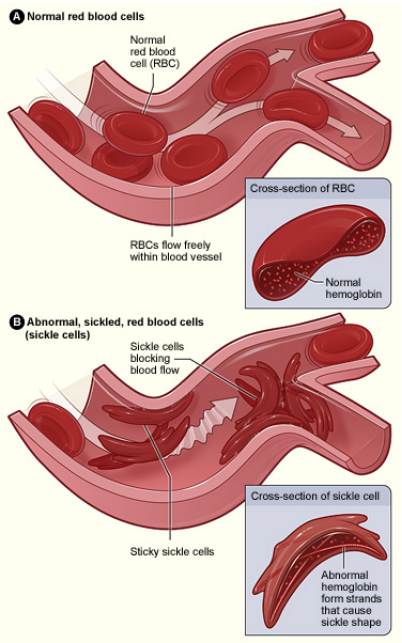

Figure 1.1: The top image shows normal RBCs flowing freely. The bottom image shows sickled RBCs in the same flow environment. [5]

Another example where flow environment has adverse effects on blood functionality is hypertension (i.e. high blood pressure). When placed in high pressure environments, it has been observed that cellular coagulation leads to the formation of thrombi [6, 7, 8, 9]. The thrombi may then detach from the endothelial wall and potentially cause an embolism. Though a minor embolism may be mild and treatable, embolisms are also a common cause of heart attack and stroke.

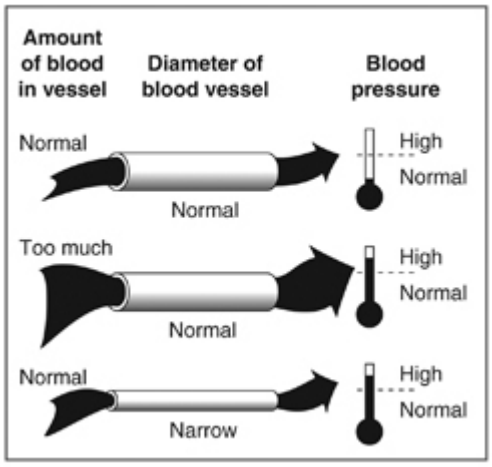

Figure 1.2: Hypertension can be caused by an effective decrease in blood vessel cross-sectional area. [10]

Therefore, a general understanding of these interactions are of great importance. Such an understanding could help treat those with existing conditions and be used to determine the need for preventative health care.

1.1 Motivation and Objective

With recent advances in computing technology, it is possible to analyze complex multi-physics systems in great detail. One such system involves biological cells interacting in a dynamic flow environment. Biological cells experience low-Reynolds number flows which bring about many phenomena not typically observed in high-Reynolds number systems. Within the model presented in this thesis, several physics are coupled to accurately represent the system and associated biological and physical phenomena; these physics include fluid dynamics, biochemical interactions, and six-degree-of-freedom motion. As part of a larger study, this model will be used in the development of a clinical tool to determine the probability of cancer metastasis in diagnosed patients [11, 12, 13].

The work presented in this paper aims to study systems of biological cells in flow via direct numerical simulation (DNS). As stated by Hoskins, et al. "In DNS methods, all flow dynamics and particle motions are discretized simultaneously, and therefore they are capable of fully resolving the physics of particle-fluid,

particle-particle, and particle-wall interactions" [12]. Though computational intensive, a DNS approach is the most physically accurate model that can be applied.

1.2 Literature Review

In the last fifty years, several attempts have been made to model flowing blood systems analytically, experimentally, and computationally. An initial motivation for these models was to gain a better knowledge of cell-to-cell adhesion and the strength of the adhesion bonds.

An early theoretical framework explored the adhesion properties of cell-cell and cell-surface systems by assuming bond formation is reversible and occurs only between specific molecules [14]. Through these assumptions it was possible to attain bond formation and breakage rates based on the system properties, including solution reaction rates and reactant diffusion rates. This framework also included a proposed bond lifetime. This lifetime was based upon the natural frequency of the system, bond energy, structural integrity, and local temperature.

A subsequent framework sought to couple bond dynamics with cellular motion [15]. This was achieved by modeling the system as a cantilever with springs fixed along the bottom of the cantilever. The opposite ends of these springs were adhered to a fixed surface using a semi-permanent, reversible adhesive (e.g. magnet on a magnetic surface). With the springs serving as the molecular bonds, the tension along the cantilever, angle of the free extremity, and spring distribution are varied in an attempt to mimic a rolling cell. For every system configuration, there is a critical tension that will result in zero rolling velocity. This assumption was tested numerically and found to exist. The parameters were then varied to understand their effects on the temporal evolution of the rolling velocity. The model showed that every configuration reached a steady state solution and was not highly sensitive to bond density at the steady state. Intuitively, these results seem valid in a flow system free of large random perturbation. Assuming the system is subject to only steady hydrodynamic forces and deterministic bond formation and breakage, the system should reach a steady state solution as time approaches infinity.

Soon thereafter, studies were conducted to explore the effect of hydrodynamic shear on the bond formation and aggregation of particles, measured as capture

efficiency [16]. Approximating a cell as a rigid sphere, the trajectory due solely to the hydrodynamic forces was calculated for a range of flow rates. These cells were then assumed to follow this exact trajectory. In the vicinity of a given location along this trajectory, another rigid spherical cell was placed. The cellular interaction was calculated using a deterministic bond formation model and assumes the moving cell would not deviate from the prescribed trajectory. This study was able to show that bond formation and capture efficiency is nearly constant along a range of shear rates until reaching a critical value. Upon reaching this critical shear rate, the capture efficiency decreases substantially. While this study employed a highly simplified system, it serves to reveal several qualitative aspects of these systems, one of which being the concept of a critical shear rate.

One study, reverting to the probabilistic bond algorithm [14], modeled the cell with several of its physical structures [17]. Though assuming a rigid body, this study did not assume the cell was a smooth sphere and allowed the bond dynamics to influence the body trajectory. By calculating the applied force due to bond formation and breakage, these bond forces were included in the calculation of the six degree-of-freedom (6DOF) motion equations. Introducing the influence of the bond dynamics into the calculation of the rigid body motions, the temporal evolution of the cell velocity was observed to be erratic and chaotic. However, it is this chaotic behavior that speaks to the stochastic nature of this system. By taking a probabilistic approach to the bond dynamics, the bonds will form and break at random, causing abrupt and impulsive motion. The velocity plot showed random perturbations around a mean value. For a given system configuration, this mean value is the velocity of the body in an unperturbed flow without the influence of bond formation and breaking.

A later study took a Lattice-Boltzmann approach to compute the forces on various cells in a "virtual blood vessel" [18]. This was a two-dimensional (2D) model in which the center of mass for every body is assumed to be in the same plane. Though the model is 2D and assumes rigid bodies, the model uses the probabilistic bond dynamics model previously mentioned. In an effort to better visualize their results, the erratic force profile was smoothed by plotting the temporal evolution of a moving average of the force acting on each body. Three flow configurations were explored computationally in this study. The first of the three configurations

was a leukocyte rolling in the virtual blood vessel in the absence of red blood cells. In this first case, each of the force profiles showed the force was relatively constant throughout the simulation. This can be attributed to the steady hydrodynamic force acting on the body. Although there are biochemical interactions between the leukocyte and the endothelium, the force is smooth due to a fluid flow field free of large perturbations.

The second case was a direct collision between a red blood cell and rolling leukocyte. In this case, the force profiles showed noticeable changes in the force between cell collision and detachment. This drastic change can be attributed to perturbations in the flow field caused by the collision, as well as surface interactions between the red blood cell and leukocyte as that will affect the trajectory and accelerations of the rolling leukocyte.

The third, and final, case of the study involved the fly-by of a red blood cell near the rolling leukocyte. The motivation for this last case is to observe the leukocyte force profiles when a red blood cell is relatively near but not close enough to interact biochemically. The flow perturbations caused a noticeable change in the force profiles, but none of these changes reached the magnitude of the direct collision.

In the last decade, several research groups have begun to use available high performance computing (HPC) resources in an effort to create higher-fidelity models.

One such study involved the modeling and simulation of a deforming leukocyte under shearing flow conditions [19]. In an effort to model a leukocyte, a spherical cell was placed near the endothelium wall. The cell was assumed to be firmly adhered to the adjacent wall, and allowed to deform. This deformation was caused solely by the hydrodynamic force and later compared to experimental observations of deformed leukocytes near vessel walls. The newly deformed shapes showed several qualitative similarities to experimentally observed deformations, including the increase of contact area between the cell and vessel wall.

Next, the leukocyte was placed near the endothelium wall without any adhesion constraints. The leukocyte began interacting with the endothelium while rolling due to the system hydrodynamics. This setup was run multiple times with varying stiffness of the leukocyte, as well as various flow shear rates. In these cases, the

trajectory, number of bonds formed and bond lifetime were recorded and compared among each of the simulation runs. It was found that flow shear rate and average rolling velocity were directly proportional, as expected. But interestingly, it was found that average rolling velocity increased with body stiffness. This correlation was attributed to the decrease in contact area, and thus a decrease in total possible bonds, of a less deformable cell. This study combined many of the underlying physics necessary in adequately modeling biological cells in dynamic flow systems, but lacked the ability to study cell-to-cell interaction.

Another study, conducted at Penn State, sought to develop a cell-to-cell interaction model that coupled the flow, structure, and biochemistry physics [11, 12]. A surface tracking method was utilized to allow for the calculation of individual bonds probabilities, as opposed to empirical "bulk" bond probabilities. One of the primary motives of this study was to develop a model to be used in the medical analysis of cancer transport, metastasis, and extravasation.

This model assumed rigid leukocytes, firmly adhered to the vessel wall, interacting with a cancer cell away from the cell wall. As the cancer cell approached the leukocyte, bond formation and breakage was calculated. The bonds themselves were modeled as Hookean springs which allowed bond force and breaking rate to be simply calculated as a function of bond length. These bond forces were then included in the 6DOF motion calculations and can be seen, as in previous studies, to directly influence the trajectory of the moving body. These forces are also incorporated into the solid mechanics equation and influence the deformation of the body.

A very interesting approach in this study was to solve the transient cellular interaction problem in a quasi-steady fashion. Given the lack of time dependency that arises in low-Reynolds number flows, a transient solution can be achieved through the computation of a series of steady state solutions [20]. Unfortunately, one large limitation of this study was its method of bond calculation. Although using the newly developed surface tracking algorithm, the algorithm was partly based on empirical data and did not allow for a realistic number of bonds. However, this model is vastly improved through implementation of the ideas presented in this thesis.

In addition to the bond limitation, the model had a very long runtime; a full

simulation had a wall clock time of 325 hours on four Intel quad-core processors. This runtime was due, in large measure, to an inefficient coupling of the various physics. The study mentions the use of several different programs to solve all of the physics, which may prove to be an area of great improvement. One obstacle in using various stand-alone software suites is the inherent start-up and close-down times required to initialize and exit each of the programs. While these times may seem insignificant to the average user, in high-fidelity modeling those times can serve as extreme performance bottlenecks. Therefore, lowering these forms of computational downtime is also an area to be explored.

Accordingly, the model proposed in this paper builds upon the work of Hoskins, et al. [11, 12], while incorporating many of the findings of previous studies. Specifically, three contributions are presented: 1) stability analysis of the coupled 6DOF-hydrodynamics system, leading to an efficient adaptive timestep specification, 2) introduction of the immersed boundary method (IBM), wherein arbitrary cell shapes are meshed internally, to enable exact inertial property evaluations, and 3) generalization to an n-body system including a novel cyclic method for reintroducing cells on periodic boundaries to enable statistical stationarity.

This thesis is organized as follows: Chapter 2 introduces the governing equations of all systems being modeled and the coupling of each system with each other. Chapter 3 describes the computational implementation of the modeling tool described in this thesis, with sections devoted to stability analysis of our blood flow model, the immersed boundary method, and generalizations of various algorithms used. Chapter 4 presents verification and validation of the tool and shows results from "arbitrarily many cell" simulations. Chapter 5 gives the research conclusion and describes future work to be completed.

Chapter 2

Theoretical Formulation

The work presented in this thesis seeks to study a coupled multi-physics system. Accordingly, each of the component physical systems must be modeled and appropriately coupled with one another. The governing equations for each of the systems (fluid mechanics, surface biochemistry, and 6DOF motion) are included here. The coupling of these systems is also explained.

2.1 Governing Equations

2.1.1 Fluid Dynamics

The flow system being modeled is comprised of a highly viscous, Newtonian fluid. The fluid motion has a low Mach number and can be approximated as being incompressible. As justified below, the flow is also approximated as being quasi-steady. Therefore, it is appropriate that the flow be governed by the steady, incompressible Navier-Stokes and continuity equations.

$$\rho u_i \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \rho g_i + \frac{\partial}{\partial x_j}\tau_{ij} \tag{2.1}$$

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{2.2}$$

For a Newtonian fluid with uniform viscosity τ_{ij} is defined as,

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{2.3}$$

In Equations 2.1 and 2.2, ρ is the fluid density, μ is the fluid viscosity, u_i is the velocity vector, x_i is the spatial coordinate vector, g_i is the gravity vector, p is the pressure, and τ_{ij} is the stress tensor.

Non-dimensionalizing the Navier-Stokes yields

$$Re \left(\frac{\partial \mathbf{u}^*}{\partial t^*} + \mathbf{u}^* \cdot \nabla \mathbf{u}^* \right) = -\nabla p^* + \nabla^2 \mathbf{u}^* \tag{2.4}$$

where the non-dimensional variables are defined as

$$x^* = \frac{x}{L} \tag{2.5}$$

$$y^* = \frac{y}{L} \tag{2.6}$$

$$z^* = \frac{z}{L} \tag{2.7}$$

$$u^* = \frac{u}{U} \tag{2.8}$$

$$t^* = \frac{t}{L/U} \tag{2.9}$$

$$p^* = \frac{pL}{\mu U} \tag{2.10}$$

In the stokes flow regime $Re << 1$. Therefore, an order-of-magnitude analysis show the terms on the left side of Equation 2.4 are negligible compared to the terms on the right side. This allows the equation to be simplified to

$$\nabla p = \mu \nabla^2 \mathbf{u} \tag{2.11}$$

Equation 2.11 is the momentum governing equation for flow in the Stokes regime. An important observation to be made in this equation is the absence of the time derivative. Systems in the Stokes flow regime are quasi-steady, mean-

ing transient cases can be modeled as sequential steady-state solutions and removes the need for mesh topology continuity. Therefore, different computational grids can be used at each instance in time and domains can be remeshed as needed to avoid skewed elements.

Throughout this work, the steady, incompressible Navier-Stokes and continuity equations (Equations 2.1 and 2.2) are used as the governing equations of the flow system. In practice, the solutions of Equation 2.1 and Equation 2.11 are equivalent in flow systems where $Re << 1$. The fluid governing equations are solved using an in-house CFD code, as described in Chapter 3.

2.1.2 Surface Biochemistry

Consistent with many of the studies found in the literature, the surface biochemistry approach used in this work is a variant of the probabilistic biochemical interaction model developed and presented by the author and his research group in previous publications [13, 21].

The bond formation and breakage probabilities are governed by a coupling of the second law of thermodynamics and equilibrium conditions for the molecules. However, the model was extended to calculate local bond probabilities [21]. In previous work [11, 12, 14, 15], the bonds were modeled as a global phenomenon. While a global approach may capture general trends of the surface biochemistry, a localized approach allows for a better understanding of bond dynamics and their effect on cellular interactions.

Localized bond formation and breakage probabilities are calculated for each adhesion molecule on every cell in the system. The association rate, k_{on}, and dissociation rate, k_{off}, are calculated as,

$$k_{on} = k_{on}^{0} n_L A_L exp\left(\frac{-s_{ts}(d-\lambda)^2}{2k_b T}\right) \tag{2.12}$$

$$k_{off} = k_{off}^{0} exp\left(\frac{(s-s_{ts})(d-\lambda)^2}{2k_b T}\right) \tag{2.13}$$

where k_{on}^{0} is the association rate at equilibrium, k_{off}^{0} is the dissociation rate at equilibrium, n_L is the molecule surface density, s is the bond spring constant, s_{ts} is

the bond spring constant during the transition state, d is the separation distance of the two molecules, λ is the equilibrium spring length, T is the local temperature, and k_b is Boltzmann's constant.

However, the value of k_{on} must be corrected as,

$$k_{on,global} = [\sum_{faces}(n_L A_L)]k_{on}^0 exp\left(\frac{-s_{ts}(d_{centroid}-\lambda)^2}{2k_bT}\right) \tag{2.14}$$

$$k_{on,average} = \frac{\sum\limits_{faces} k_{on}}{\text{number of faces}} \tag{2.15}$$

$$k_{on,corrected} = k_{on}\frac{k_{on,global}}{k_{on,average}} \tag{2.16}$$

where $d_{centroid}$ is the distance between the centroids of the two cells.

For each molecule pair, the probability of bond formation is calculated as,

$$P = 1 - exp(-k_{on,corrected} * \Delta t) \tag{2.17}$$

where Δt is the elapsed simulation time since the previous calculation. A random number is then generated and a bond is formed if the calculated probability is greater than the random number. Similarly, bond breakage is calculated as,

$$P = 1 - exp(-k_{off} * \Delta t) \tag{2.18}$$

To simplify this model for validation, bonds were only allowed to occur between LFA-1 and ICAM-1 molecules. An idealized representation of this system can be seen in Figure 2.1.

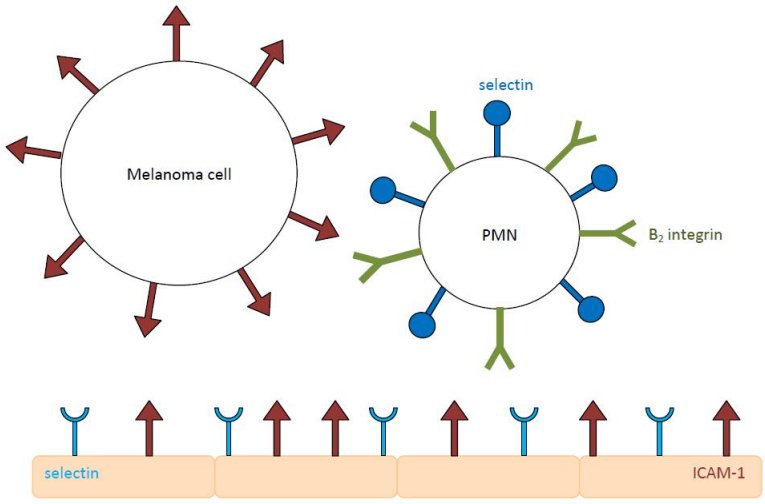

Figure 2.1: ICAM-1 molecules on the surface of melanoma cells interact with β-2 integrins on the surface of PMNs. In this study, the only β-2 integrin allowed to interact biochemically was LFA-1. Once adhered to a PMN, melanoma cells can exit the bloodstream once PMNs begin interacting with the endothelial cells. [13, 21]

2.1.3 Six Degree-of-Freedom Motion

The bodies traveling through the flow field are treated as rigid bodies that move with six degrees of freedom; each of the bodies experiences three components of translation and rotation, respectively. This motion can be described as an extension of Newton's second law of motion,

$$\frac{d^2 x_i}{dt^2} = \frac{F_i}{m} \tag{2.19}$$

$$\frac{d^2 \theta_i}{dt^2} = \frac{T_i}{I} \tag{2.20}$$

where x_i is the linear displacement vector, θ_i is the angular displacement vector about the body's centroid, F_i is the sum of all forces applied at the centroid, T_i is the torque applied to the centroid, m is the mass of the body, and I is the principal moment of inertia corresponding to the i^{th} principal axis.

To accommodate bodies of arbitrary geometry, the principal moments of inertia are calculated as,

$$I_C = \int_V \rho(r_i) r_i^2 dV \tag{2.21}$$

for each principal axis where r_i is the radius vector to a point in the body from the principle axis passing through the body centroid C, and $\rho(r_i)$ is the mass density at each point r_i.

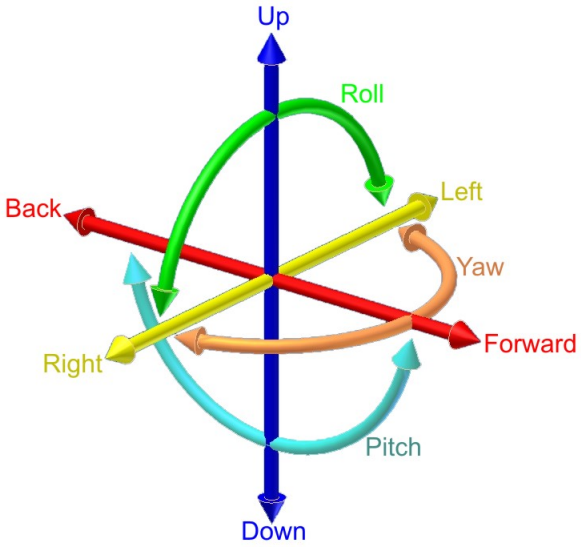

Figure 2.2: The six degrees-of-freedom [22]

The components of 6DOF motion can be seen in Figure 2.2. An object with 6DOF may experience motion comprised of any combination of these components. Care must be taken when calculating the overall motion. While translational components of motion can be arbitrarily summed and applied, rotational components of motion must be applied in the correct order.

2.2 Coupling of Dynamic Systems

In an effort to better understand cellular interactions with an emphasis on cancer transport, it seems intuitive to use the motion equations as a coupling mechanism. Both the fluid mechanics and the surface biochemistry of each cell imparts some measurable effects on every other cell in its vicinity. Thus, to couple these dynamic systems, these effects are calculated and included in the motion calculations for each of the cells.

2.2.1 Hydrodynamics

The forces and torques that determine cell motion are partly due to the fluid mechanics. Any body moving through a real fluid experiences pressure and viscous forces on the surface of the body. The distribution of these forces can have a significant effect on cell trajectory.

Given the solution to the flow field, the forces and torques acting on a body can be solved as,

$$F_{i,fluid} = \int_A [(-p\delta_{ij} + \tau_{ij})\hat{n}_j]dA \tag{2.22}$$

$$T_{i,fluid} = \int_A \{r_i \times [(-p\delta_{ij} + \tau_{ij})\hat{n}_j]\}dA \tag{2.23}$$

where $\hat{n}_j$ is the unit normal of the j^{th} face on the body, r_i is the radius vector from face j to the centroid of the body, and δ_{ij} is the Kronecker delta.

2.2.2 Biochemistry

Similarly, the surface biochemistry has associated forces and torques. When bond formation occurs, equal and opposite forces are exerted along the surface-to-surface line of action. Therefore, the bond forces are applied at the locations where the line of action intersects with the surfaces of each cell.

All bond forces on a given cell are then applied at the cell's centroid with the resulting torque also being applied about the centroid. The strength of the forces and torques are computed based on the approximation that bonds behave as Hookean springs. The force and torque acting on the centroid can be expressed as,

$$f_{j,bonds} = s(d - \lambda)\hat{e}_j \tag{2.24}$$

$$F_{i,bonds} = \sum_{bonds} [(f_{j,bonds} \bullet \hat{n}_i)\hat{n}_i] \tag{2.25}$$

$$T_{i,bonds} = \sum_{bonds} \Big\{ r_i \times [(f_{j,bonds} \bullet \hat{n}_i)\hat{n}_i] \Big\} \tag{2.26}$$

where s is the spring constant, d is the distance between the two molecules, $\hat{n}_i$ is the unit normal along the i^{th} coordinate axis, and $\hat{e}_j$ is the unit normal in the along the line of action of the bond.

2.2.3 Modeling of Surface Roughness

Since the cells are approximated as having smooth surfaces, sub-grid modeling is used to account for the micro-structures on the surfaces of the cells. As seen in Figure 2.3, these micro-structures can be fairly complex and are unique to each cell type. Therefore, all biological cells being simulated must maintain a minimum separation distance from any other body. The minimum separation distance corresponds to the surface roughness. Enforcement of this separation distance is accomplished using a fictitious repulsion force.

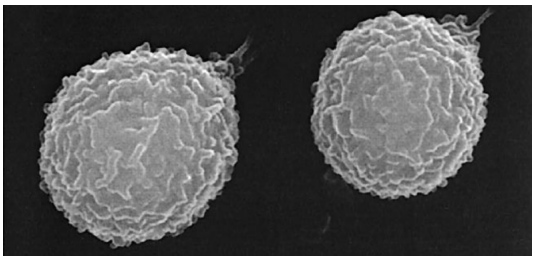

Figure 2.3: Cell surfaces contain many complex structures which are modeled using a sub-grid fictitious repulsion force. [11, 23]

Figure 2.4 shows two cells in close proximity. The dashed blue curve represents the surface of the discretized body used in the simulation. However, this minimum separation distance must accounts for the cells' unresolved surface features. The length of these features are incorporated into the minimum separation distance, referred to in Figure 2.4 as the critical distance.

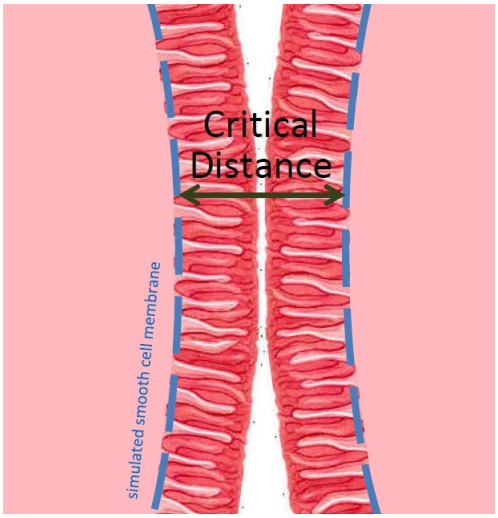

Figure 2.4: A fictitious repulsion force models the presence of the microvilli. This force ensures the cells are never closer than the critical distance. [13]. Adapted from [24]

As explained in [11, 12, 13, 21], the fictitious force used is solely a function of surface separation distance and can be applied to multiple bodies with varying surface roughness characteristics. The repulsion force is modeled as a non-linear spring force applied at the surface of each cell. The repulsive force is defined as

$$f_{i,rep} = [ad + bd^3]\hat{e}_i \tag{2.27}$$

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{2.28}$$

$$F_{i,rep} = \sum_{faces} f_{i,rep} \tag{2.29}$$

$$T_{i,rep} = \sum_{faces} \left\{ r_i \times f_{i,rep} \right\} \tag{2.30}$$

where a and b are spring constants representing the forces exerted by the surface structures, d is the distance function evaluated between the two faces, and $\hat{e}$ is the unit normal vector along the line of action between the two computational faces. For the TC-PMN cell pairs being simulated in this paper, the spring constants

have been set to $a = -110 \times 10^{-6}\text{N/m}$ and $b = 600 \times 10^{6}\text{N/m}^3$ to maintain a minimum separation of 0.3 μm. In this work, the fictitious force is calculated and applied at separation distances less than $\epsilon = 1.2\mu$m.

Chapter 3

Computational Implementation

One of the contributions of this thesis is the implementation of the multi-physics formulations onto a computational platform. Such a platform allows the formulations to be used as either a predictive or diagnostic tool. Implementation on a computational platform is consistent with the long-term goal of this work; the development of a predictive tool for use in personalized health care of cancer patients.

The steady, incompressible Navier-Stokes and continuity equations governing the flow system are solved using NPHASE-PSU, an in-house CFD code. NPHASE-PSU employs a segregated pressure based methodology using a colocated variable arrangement and a lagged coefficient linearization. A finite volume method is used to discretize the momentum equation (Equation 2.1), and continuity (Equation 2.2) is introduced via a pressure corrector equation, based on the SIMPLE-C method [25].

Given the localized approach to solve the surface biochemistry, the biochemistry equations are solved within NPHASE-PSU as a subroutine. Solving these equations within NPHASE-PSU allows for the exploitation of existing data structures. The biochemistry routine is able to access the geometrical properties of the discretized model being analyzed by NPHASE-PSU; one such property is the centroid location of discretized faces. The routine assembles a list containing all faces on the surface of a biological cell in the flow domain, along with the face centroid and outward unit normal of the face. One such list is then created for each biological cell in the flow domain. Once created, these lists are used to cal-

culate face separation distances in Equation 2.28 and the probabilities for bond formation and breakage.

After solving the bond dynamics, it is possible to incorporate the effects of these bonds into the rigid body motion calculations. For each existing bond, the bond force is applied at the centroid of the face to which the bond is attached. The bond forces are expressed in component form, allowing for the motion to be computed in each of the three spatial dimensions. Then, the bond forces are used to calculate the corresponding torque acting on the biological cell.

Before calculating the rigid body motion, it is necessary to compute the force and torque acting on the cells due to the hydrodynamics. Essentially, the previous method is repeated for the pressure and viscous effects on each biological cell.

The forces and torques due to hydrodynamics and bond dynamics are summed and then used in the motion calculation. After each of the components of force and torque are known, Equations 2.19 and 2.20 are solved for the linear displacement vector, x_i, and the angular displacement vector about the body's centroid, θ_i.

While this approach is seemingly straightforward, there are several nuances that must be addressed during implementation.

3.1 Numerical Stability of ODEs

A main issue to be addressed in the implementation of this model is the stability of the solution technique. As previously shown, Equations 2.19 and 2.20 are ordinary differential equations (ODEs). Given some starting condition, this problem can be described as an ODE initial-value problem (IVP). Such a problem is solved using a numerical scheme that incrementally advances the solution with respect to the variable of differentiation.

The most basic of schemes for solving this type of problem is the explicit Euler method. This method is first-order accurate, meaning the error of each incremental advancement is proportional to the square of the increment size. Also, the global/total error is proportional to the increment size.

Formally, the explicit Euler method is expressed as,

$$y(t + \Delta t) = y(t) + \Delta t \frac{dy(t)}{dt} \tag{3.1}$$

where y is the variable to be solved at each discrete point in time, t is the time, and Δt is the time increment (or timestep).

Any ODE numerical solution scheme can be evaluated by calculating the stability of the method. Numerical stability can be described as a measure of error propagation as the scheme advances the solution. A scheme with large amounts of error propagation is considered to have low stability or may be unstable.

To test the stability of the explicit Euler method for the quasi-steady simulation of the present dynamic flow systems, a simplified ODE was developed and solved using this method. The ODE used was obtained by enforcing various assumptions on the original system. While this representative ODE will not capture all of the system's dynamics, it is useful in developing a fundamental understanding of the solution stability when applying the explicit Euler method.

The first assumption is that drag on the body is due solely to the hydrodynamic forcing. It was also assumed that the cells were smooth, rigid spheres for all time and that the cells were restricted to one-dimensional translation. Given these assumptions, along with the knowledge that these systems have low Reynolds numbers ($Re < 1$), it is possible to obtain an expression for F_i in terms of $\frac{dx_i}{dt}$.

For $Re < 1$,

$$C_{Drag} \equiv \frac{F_{Drag}}{\frac{1}{2}\rho u^2 A} = \frac{24}{Re} \tag{3.2}$$

$$Re = \frac{\rho u D}{\mu} = \frac{2\rho u r}{\mu} \tag{3.3}$$

$$\Rightarrow F_{Drag} \approx 6\pi\mu r u \tag{3.4}$$

where u is a reference velocity expressed as $u = u_{freestream} - u_{body}$.

Substituting Equation 3.4 into Equation 2.19 yields,

$$\frac{du_{body}}{dt} = \frac{F_{Drag}}{m} \approx \frac{6\pi\mu r u}{m} \tag{3.5}$$

$$\text{where } m = \rho V \text{ and V is the cell volume} \tag{3.6}$$

Given that $u_{freestream}$ is a constant, Equation 3.5 can be written as

$$\frac{du}{dt} \approx -Au \tag{3.7}$$

$$\text{where } A = \frac{6\pi\mu r}{\rho V} \tag{3.8}$$

Equation 3.7 is the simplified ODE used to test the stability of any numerical scheme for this application. The equation used to advance the solution through time can now be expressed by substituting Equation 3.7 into Equation 3.1.

$$u^{t+\Delta t} = u^t - \Delta t A u^t \tag{3.9}$$

Physically, convergence in this type of problem is defined as u_{body} reaching, then maintaining, the terminal velocity of the given body. This terminal velocity is reached when $F_{Drag} = 0$. Given the approximation of F_{Drag} in Equation 3.4, it is easily seen that $F_{Drag} = 0$ when $u = 0$. Therefore, in the context of advancing the solution in time, the model is converging iff $|u^{t+\Delta t}/u^t| \leq 1$. Further manipulation gives,

$$\left|\frac{u^{t+\Delta t}}{u^t}\right| = |1 - \Delta t A| \leq 1 \tag{3.10}$$

$$\therefore \Delta t_{max} = \frac{2}{A} \tag{3.11}$$

According to the stability analysis, solving this problem with the explicit Euler method will converge for $\Delta t \leq \frac{2}{A}$ where $A = \frac{6\pi\mu r}{\rho V}$. For this specific test case, $\mu = 0.001\frac{\text{kg}}{\text{ms}}$, $r = 8\mu\text{m}$, $\rho = 1087\frac{\text{kg}}{\text{m}^3}$, and $V = \frac{4}{3}\pi r^3$. Substituting the values into Equation 3.11 yields

$$\Delta t_{max} = \frac{\rho V}{3\pi\mu r} \approx 30\mu s \tag{3.12}$$

This stability analysis was also performed empirically by solving Equation 3.9, where Au is the Stokes drag force per unit mass, and marching the solution in time using the Explicit Euler method. The temporal evolution of u in Equation 3.9 was obtained several times, with Δt varying in each evolution. The Δt_{max} was the largest value of Δt that converged to a solution. The results of this analysis

can be seen in Figure 3.1, where reference velocity is the difference between u and u_{terminal}.

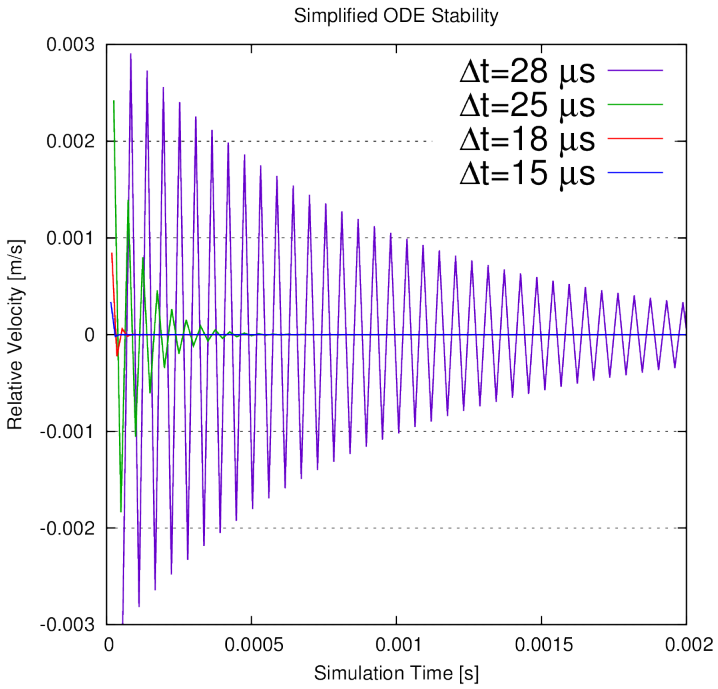

Figure 3.1: Empirical test showing stability up to 28 μs

Empirically, Δt_{max} was found to be $28\mu s$. However, due to the slow convergence with this timestep, it is not advantageous to use this value during dynamic simulations. As per the results shown in Figure 3.1, Δt of this system should be less than $20\mu s$ to accurately capture system dynamics and ensure stability.

To test the empirically found stability limit, the flow system was simulated using NPHASE-PSU. A sphere with zero-velocity was placed along the centerline of a fully-developed flow. This simulation was run several times, using each of the four timesteps found in Figure 3.1. However, the stability limit of the 3D simulation was found to be lower than the limit found by solving Equation 3.9. The results of these simulations can be found in Figure 3.2.

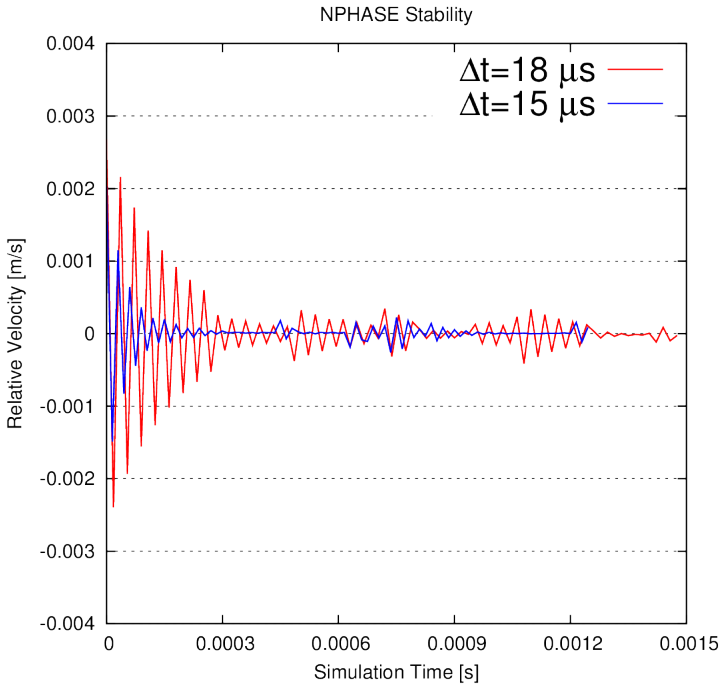

Figure 3.2: Results of 3D terminal velocity simulation showing stability up to 18 μs

In the 3D terminal velocity simulation, the maximum converging timestep was found to be 18 μs. This lower stability limit is to be expected, as the 3D simulation solves Equations 2.1 and 2.2 to compute the drag force on the sphere then solves Equations 2.19, 2.20, and 2.21 using the Explicit Euler method and the computed value of drag. While solving the set of equations previously mentioned allows for fewer assumptions, this approach leads to reduced stability of the system.

For greater stability, it is possible to implement an explicit numerical scheme with higher order accuracy. Such schemes include the midpoint method, Heun's method, and the classic Runge-Kutta method (RK4). Empirical stability analyses were also performed on these methods. These results can be found in Table 3.1.

Table 3.1: Empirical Stability Analysis

	Maximum Δt	Local Truncation Error	Global Error
Explicit Euler Method	28 μs	$O(\Delta t^2)$	$O(\Delta t)$
Heun's Method	28 μs	$O(\Delta t^3)$	$O(\Delta t^2)$
Midpoint Method	28 μs	$O(\Delta t^3)$	$O(\Delta t^2)$
RK4 Method	39 μs	$O(\Delta t^5)$	$O(\Delta t^4)$

However, there are some concerns when implementing any of these higher-order accurate schemes. While explicit Euler requires the ODE be evaluated once per time step, higher order methods require the ODE be evaluate multiple times per time step. This translates to a longer runtime for the overall code. Thus, for this application the increase in stability does not justify the increase in runtime.

3.2 Immersed Boundary Approach

In an effort to understand cellular interactions, the biological cells within the flow domain are discretely resolved. By allowing discrete particles to flow through the domain, it is necessary to have a structure for each of the bodies. Typically, a structure of this sort would be given solid boundary conditions. However, for reasons motivated by the future of the project, an immersed boundary approach was used to allow for more robust control of the cell bodies.

For sake of historical perspective, it is necessary to mention the immersed boundary (IB) method developed by Peskin [26]. The Peskin-IB method was proposed as an approach to couple structural deformations and fluid flow in fluid-structure interaction (FSI) simulations. The method consisted of a Eulerian (or fixed) computational mesh for the fluid flow and a Lagrangian (or conformal) mesh for the solid body. Initially developed for two-dimensional problems, the solid body would be projected onto the fixed fluid mesh as a parametric curve. The immersed body, which is now projected onto the fluid mesh, applies some forcing function of the fluid. This forcing function changes the flow field and the shape of the solid body is updated based on the resulting deformation. The process is then repeated for the duration of the simulation.

While the Peskin approach is not used in this work, the IB approach described here is motivated by the Peskin method. It was desired to develop an IB approach that could exploit the existing data structures in NPHASE-PSU. These exploitations will be important as future simulations are to include structural mechanics and mass transport across the cell membrane.

Prior to implementation of the NPHASE-IB approach, cellular surfaces were treated as solid wall boundary conditions. Figure 3.3 shows a sample mesh using the rigid wall approach, where it can be seen that the fluid domain is discretized

while the inside of the cell body is not discretized. This approach has several limitations. First, the approach assumes the body is rigid and incompressible. Second, in the context of 6DOF motion, the principal moments of inertia cannot be calculated without assuming constant density throughout the body. Third, this approach does not allow for permeability of the cellular wall. Many cellular functions involve mass transfer across the cellular membrane, a condition with cannot be specified using a solid wall boundary condition.

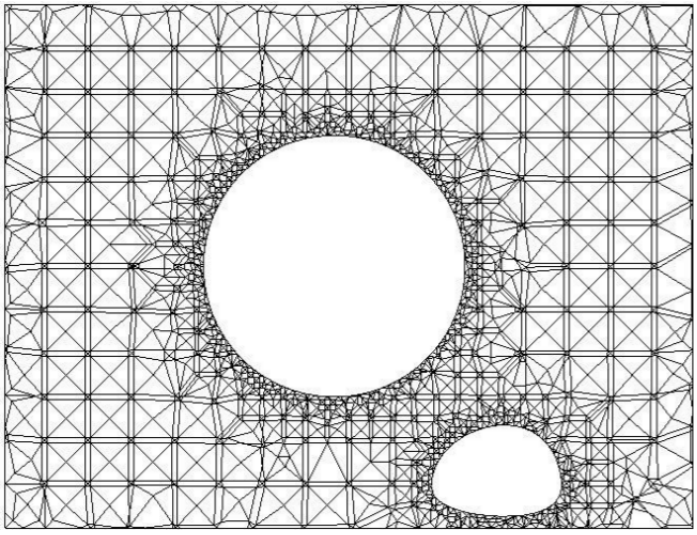

Figure 3.3: Sample mesh using the rigid wall approach. Inside of the body is not discretized.

To address these limitations, it was decided that the surface of the cell body be treated as an internal flow face with prescribed flux and velocity at the surface. Such a surface allows for a range of easily enforceable boundary conditions. Also, the solution procedure for the flow problem does not need to change because the conditions enforced on the cell surface are applied using existing data structures.

The NPHASE-IB approach was accomplished by merging two meshes, one of the discretized fluid domain and one of the discretized solid domain. In practice, this was done by first creating a surface mesh of the cell body (as is also necessary in the rigid wall approach). The flow domain was then discretized, creating a fluid mesh similar to Figure 3.3. The same surface mesh is then used to create a mesh of the discretized solid domain; namely, the inside of the cell body. Having created two volume meshes using the same surface mesh, a single volume mesh is created by

matching and merging the faces of the surface mesh. The faces belonging originally to the solid mesh are designated as "immersed faces" and treated accordingly. An example of a mesh created using this process can be seen in Figure 3.4.

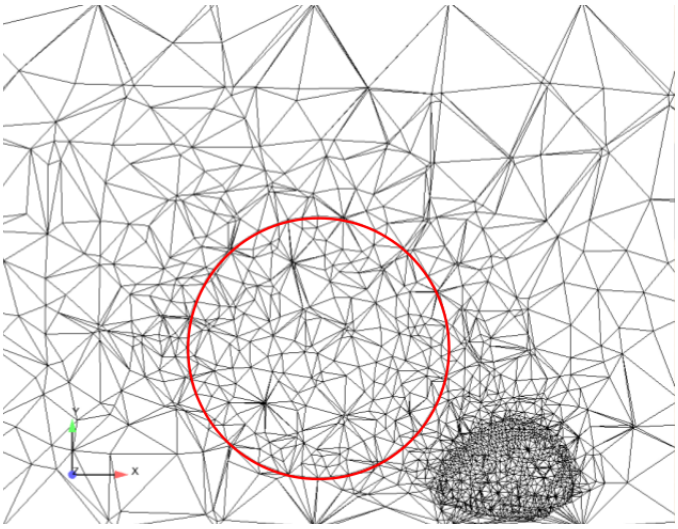

Figure 3.4: Sample mesh using the immersed boundary approach. Inside of the body is discretized and solved as internal flow elements. The red curve denotes the boundary of the cellular wall.

In the context of structural mechanics, this approach allows the surface of the body to be treated as an elastic membrane. By also meshing the inside of the body, it is possible to solve the structural mechanics as a two fluid problem separated by this membrane. Additionally, this approach enables generalized evaluation of inertial properties are shown in Section 3.4.

3.3 Generalization of Biochemistry Algorithm

To model a large flow system, a great deal of computational resources may be needed. To access all the necessary resources, many computational tools are parallelized. The act of parallelizing a code allows the software to use multiple processors to solve a single problem. Fortunately, NPHASE-PSU is fully parallel and capable of simultaneously running on an arbitrary number of processors. With this capability, the flow domain is first decomposed into n partitions (where n = Number of Processors) before being sent to the flow solver.

While the flow solver within NPHASE-PSU is capable of multi-core computation, a subroutine written within NPHASE-PSU does not inherently have that capability. In order to run the biochemistry routines in parallel it was necessary to generalize the algorithm to run on any number of processors and machines.

As previously explained, the biochemistry routine works by scanning all face pairs on the surface of the biological cells and computing the relevant data using Equations 2.12 and 2.13. Therefore, the domain must be decomposed into n sections, then the face pairs can be scanned for every pair of decomposed domains. Since NPHASE-PSU already has a decomposed domain when running in parallel, it is possible to use that same data for the biochemistry routine. Then, parallelizing the routine becomes a scheduling problem. This scheduling problem is to ensure that each section of the decomposed domain is only paired with one other section of the decomposed domain at any given time during the face-pair scan and computation.

This scheduling problem is solved by implementing a modified version of the round robin scheduling algorithm [27]. The round robin scheduling algorithm is commonly used to schedule tournaments for various competitions and is designed such that each team plays every other team at least once during the tournament. Naturally, this algorithm is well suited to parallelizing the biochemistry routine where every processor must be paired once with every other processor. After running the algorithm, every face pair will have been scanned and all relevant data will have been computed; then the information can be compiled on a single machine. This routine is modified such that every processor is also paired with itself.

Using the modified algorithm, it is necessary to run the biochemistry pairing-computing routine n times. During the first iteration, each processor is paired with itself; this means for a given section of the domain decomposition, the face pairs can only be formed with faces within that section of decomposition. For the following $n - 1$ iterations, each processor will be assigned to a pair; then a face pair must consist of one face from each of the two processor.

To ensure that every processor has paired with every other, the pairs are pivoted around an initial pair group. An example of an initial pair group can be seen in Table 3.2a. One processor is then selected to remain fixed. For every subsequent

iteration of the scheduling algorithm, the other $n-1$ processors are rotated one position. The rotation step is repeated until the pair group matches the initial pair group. An example of the entire process is shown in Table 3.2. This example assumes that NPHASE-PSU uses four processors to solve the problem. In the final three iterations, the first processor is held fixed during the rotation step.

It is important to note, this algorithm can also be used with an odd number of processors. Assuming NPHASE-PSU uses only three processors to solve the problem, a fourth dummy processor will be added to the list. This dummy processor serves as a place holder to ensure the rotations are performed correctly. Any processor paired with the dummy processor will simply idle and not perform the biochemistry routine. A psuedocode implementation of the mRRSA can be found in Algorithm 1.

Table 3.2: Scheduling Algorithm - Example Case

(a) First Iteration

1	2	3	4
1	2	3	4

(b) Second Iteration

①	2
3	4

(c) Third Iteration

①	3
4	2

(d) Fourth Iteration

①	4
2	3

Algorithm 1 Modified Round-Robin Scheduling Algorithm

```
Set n = Number of processors
Initialize temp[n]
Initialize schedule[2 * n]
// Pair group for processor val is (schedule[val], schedule[val + n])
Set totRuns = n + 1
for i:=1 to totRuns step 1 do
    if i == 1 then
        for j:=1 to n step 1 do
            temp[j] = j
            schedule[j] = j
            schedule[j + n] = j
        end for
    else
        for j:=1 to n step 1 do
            // Set all values of schedule to -1
            // -1 means processor already belongs to a pair
            schedule[j] = −1
        end for
        for j:=1 to n/2 step 1 do
            proc1 = temp[j]
            proc2 = temp[n − j + 1]
            schedule[proc1] = proc1
            schedule[proc1 + n] = proc2
        end for
        if n is even then
            Swap values of temp[1] and temp[n]
        end if
    end if
    Run Biochemistry Routine
    // Rotate list temp
    tempVal = tmp[n]
    for j:=1 to n step 1 do
        temp[n − j + 1] = temp[n − j]
    end for
    temp[1] = tempVal
end for
```

3.4 Generalization of 6DOF Algorithm

In discrete modeling of dynamic blood flow systems, it becomes necessary to accommodate various bodies of arbitrary geometries, e.g. red blood cells, white blood cells, and tumor cells. Therefore, a generalized 6DOF algorithm was needed to accommodate bodies of arbitrary geometry.

Obtaining such an algorithm requires the ability to compute the principal components of inertia for an arbitrary body. This is done by meshing the interior of the body into finite volumes, as discussed in Section 3.2, and calculating the body's moment of inertia as a discrete system of masses. The first step in calculating the inertia is computing the body centroid. The centroid is computed as,

$$C_i = \frac{\sum_{j}^{\text{interior elements}} m_j c_{ij}}{m_{total}} \tag{3.13}$$

$$m_{total} = \sum_{j}^{\text{interior elements}} m_j \tag{3.14}$$

$$m_j = \rho_j V_j \tag{3.15}$$

where m_j is the mass of interior element j, c_{ij} is the i-th coordinate of the centroid of interior element j, ρ_j is the density of the fluid in interior element j, and V_j is the volume of interior element j.

Once the body centroid has been found, it is possible to compute the principal components of inertia as

$$I_i = \sum_{j}^{\text{interior elements}} m_j r_{ij}^2 \tag{3.16}$$

$$r_{ij} = |c_{ij} - C_i| \tag{3.17}$$

After m_{total} and I_i have been calculated for each body, it is then possible to use Equations 2.19 and 2.20 to compute the motion of the body given the force and torque acting on the body.

An example of the internal meshes used to compute 6DOF properties can be

seen in Figure 3.5. The object's internal mesh elements are used to calculate its centroid and principal components of inertia.

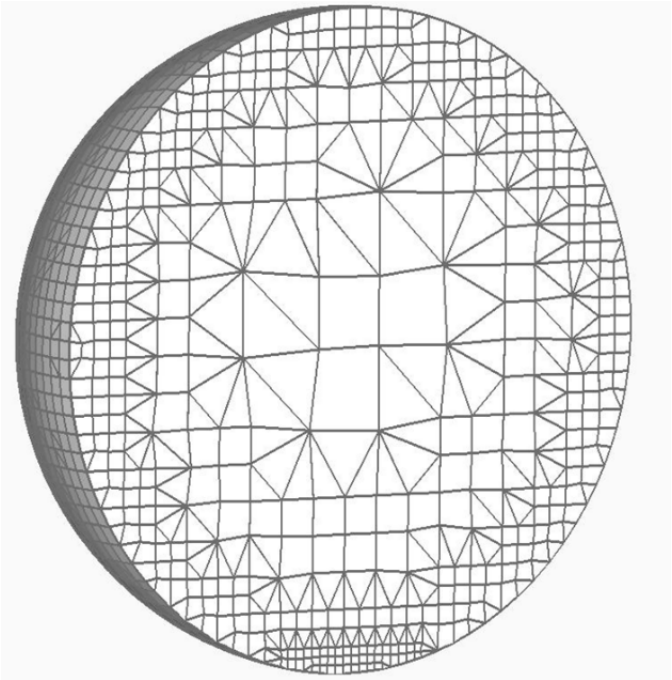

Figure 3.5: Sample mesh showing internal elements used in generalized 6DOF calculations. [11]

3.5 Generalization to n-Body Simulations

Upon implementation of the biochemistry and 6DOF algorithm, it was possible to calculate the dynamic motion and interaction for a pair of biological cells. However, the model would need to accommodate an arbitrary number of cells for use in a clinical setting. Naturally, the next step was to generalize the algorithm to handle any number of bodies in the flow domain.

This generalization was accomplished by creating an additional data structure within NPHASE-PSU to identify each body in the domain. This was done by assuming each body consisted of a fluid of unique properties. Therefore, the material identification number of the fluid flowing through the domain was set to zero and each of the bodies is assigned a unique identification number. Then given a finite

number of bodies, a subroutine was created within NPHASE-PSU to perform the biochemistry and 6DOF computations for each body.

3.5.1 Cycling of Cells through Domain

While the algorithm could handle an arbitrary number of bodies, any computer simulation is limited by the finite amount of resources available for use. Because of this, it may not be feasible to run a simulation with thousands (or millions) of blood cells. To circumvent this limitation and study systems with a large number of cells, the concept of cell-cycling was introduced.

Cell-cycling is a repositioning of cells within the flow domain to simulate a flow through a very long vessel. As a cell reaches the end of the flow domain, it is moved to the entrance of the flow domain, as seen in Figure 3.6. All body and flow properties are preserved when the cell is moved; these properties include geometry, linear velocity, angular velocity, and location relative to the endothelial wall. The body is moved by a simple changing of the centroid coordinate corresponding the flow stream-wise direction. This changing of the centroid coordinate is accommodated by adding a fictitious force to the 6DOF motion calculation of the body which results in the change of centroid location. The fictitious force used to produce the change of centroid location is only applied when the body fully enters the exit buffer zone. The exit buffer zone is the last two reference cell diameters of the domain in the streamwise direction, where the reference cell diameter is the diameter of the largest cell in the flow domain. Once in this zone, the cell is moved to the entrance buffer zone which is the first two reference cell diameters of the domain in the streamwise direction. Creating these buffer zones allows data from the domain's area of interest to be free of any influence from the computational implementation of the cell cycling.

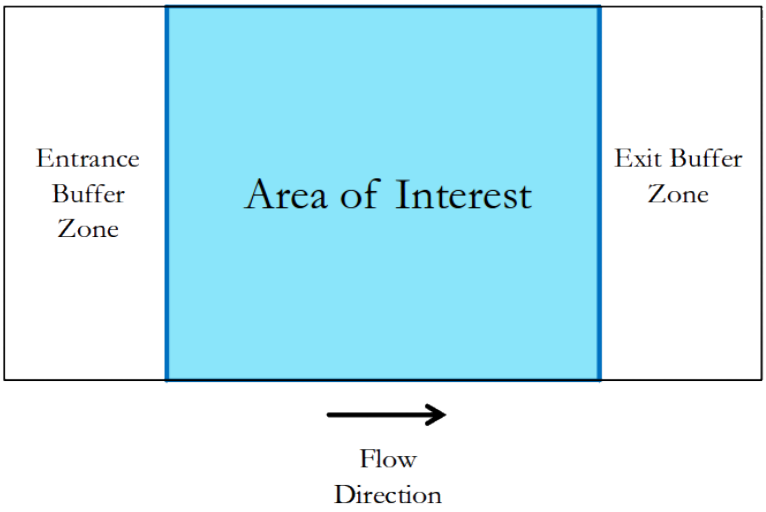

Figure 3.6: Flow domain with buffer zones

By implementing cell-cycling it is possible to significantly reduce the simulation cost of studying large blood flow systems. This technique also allows the domain to evolve to statistically stationary solutions.

Cell cycling was motivated by the idea of periodic boundary conditions often used in computational mechanics. Such boundary conditions enforce periodic conditions on domain faces within computational models to reduce computational cost of analyzing cyclic systems (e.g, gears, propellers, rotating tires, etc). Therefore, it was straightforward to extend the concept of periodic boundary conditions to cycling bodies moving within a flow domain.

3.6 Miscellaneous High Performance Computing Improvements

In addition to the on-board 6DOF calculations and the parallelization of the biochemistry module, several other improvements were made to increase computing performance.

Adaptive time-stepping was used to ensure proper temporal resolution throughout the solution. This was done by adjusting the timestep based on a deviation of the force from the mean force acting on a body. For example, bond formation

introduces a large force of the bodies involved. This increase in force results in a lowering of Δt to ensure the dynamics are being adequately captured.

Convergence criteria were also introduced to ensure proper convergence without wasting unnecessary computing cycles. During runs of NPHASE-PSU, the root-mean-square of a chosen residual error is monitored. The flow solver is assumed to have converged when the chosen residual error has reached a specified threshold.

Lastly, a meshing strategy was developed to decrease meshing time. Surface meshes for different cell types are stored as stereo-lithography (i.e., .stl) files. Domain boundaries are also meshed during pre-processing and each surface mesh is saved. Then, after computing cell motion, only volume mesh elements must be recomputed at each timestep. In addition to decreases in runtime, this approach also allows for the tracking of individual bonds since surface topology is preserved through time.

Collectively, these improvements have significantly reduced simulation runtime.

Chapter 4

Verification, Validation, and Results

Upon implementation of the formulations previously discussed, various test cases were developed for use in verification and validation. These cases serve to demonstrate the capabilities of the formulations and ensure they are behaving as expected. By separating the test cases into two categories, it is possible to draw similarities and distinctions.

The two categories are cell-pair simulations and arbitrarily many cell simulations.

4.1 Cell-Pair Simulations

Having verified the 6DOF motion module is working correctly, the next step is to ensure the biochemistry algorithm is behaving as expected. This is accomplished by modeling two biological cells in a flow domain. The two cells are placed in a Couette flow similar to that used in the single cell simulations. The first body, a rigid, spherical tumor cell is initially placed in the computational domain, such that is will move freely through the flow domain. The second cell, a white blood cell, is deformed as described in previous work [11, 12, 13, 21] and is fixed on the bottom wall of the domain.

4.1.1 Repulsion Module

Before exploring the capabilities of the biochemistry module, it was necessary to test the repulsion model. As discussed earlier, this model serves to prevent the bodies from becoming non-physically close.

This is done by computing the forces and torques associated with modeling the unresolved surface features using Equations 2.29 and 2.30, and including these forces and torques in the 6DOF motion calculations as described in Section 2.2.3. The two cells placed in the flow are allowed to interact via the repulsion force. A few cases were run varying the critical distance of minimum separation, ϵ (refer to Section 2.2.3). The system constants used in these simulations can be found in Table 4.1.

Table 4.1: Repulsion Model - System Constants

Parameter	Value
λ [μm]	0.05
$\dot{\gamma}$ [s^{-1}]	150
Domain Size [μm]:	
X	60
Y	32
Z	42
Tumor Cell Initial Centroid [μm]:	
X	18
Y	10
Z	21
PMN Initial Centroid [μm]:	
X	30
Y	2.5
Z	21

As seen in Figure 4.1, there is an observable difference in the trajectories of each case; namely, a larger critical distance corresponds to a larger separation between the bodies. Aside from separation distance, each trajectory is nearly identical. The high level of similarity between each of the trajectories is expected as the system is deterministic without the biochemistry module. Varying the critical distance can be considered as adjusting the effective length of the unresolved surface features, e.g. microvilli (see Figures 2.3 and 2.4).

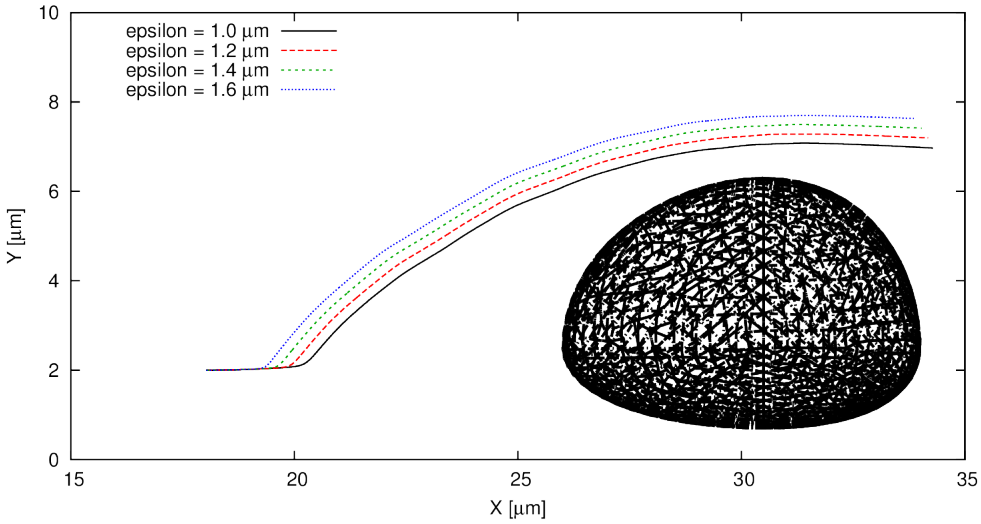

Figure 4.1: Cell trajectories for each of the cases when adjusting repulsion parameters. Trajectory shows point closest to white blood cell at each timestep.

4.1.2 Biochemistry Module

Several more cases were then created to explore the capability of the biochemistry module. These cases were placed in two subcategories. The first category varied the spring constants used in equations 2.12 and 2.13 to calculate affinity of bond formation and breakage, while the second category varied the relative position of the bodies and shear rate of the flow.

4.1.2.1 Parameterization of Adhesion Model

With the critical distance fixed, the biochemistry parameters were systematically varied to observe any trends that may arise. Specifically, the spring constants of the modeled bonds are varied to cause an effective increase or decrease in bond strength. It is to be expected that high-strength bonds have a greater impact on trajectory than their low-strength counterparts. To simplify the analysis, bonds were only allowed to form between LFA-1 and ICAM-1 molecules. The system constants chosen for this analysis can be found in Tables 4.2 and 4.3.

Table 4.2: Parameterization of Adhesion Model - System Constants

Parameter	Value
k^0_{off} [s^{-1}]	0.3
k^0_{on} [s^{-1}]	3×10^{-3}
ϵ [μm]	1.2
λ [μm]	0.05
$\dot{\gamma}$ [s^{-1}]	150
Domain Size [μm]:	
X	60
Y	32
Z	42
Tumor Cell Initial Centroid [μm]:	
X	18
Y	10
Z	21
PMN Initial Centroid [μm]:	
X	30
Y	2.5
Z	21

Table 4.3: Parameterization of Adhesion Model - Variable Parameters

	ADH1	ADH3	ADH4	ADH5	ADH6
s [N/m]	2×10^{-3}	2×10^{-4}	1×10^{-3}	2×10^{-5}	1×10^{-4}
s_{ts} [N/m]	1×10^{-3}	1×10^{-4}	5×10^{-4}	1×10^{-5}	5×10^{-5}

As seen in Figure 4.2, all of the cases result in the capture of the tumor cell. Each of these cases was run for 3000 timesteps and the capture criterion for the tumor cell was when the value of velocity in the flow direction reached zero (as shown in Figure 4.3). However, the differences in each case can be seen by plotting velocity with respect to centroid location. Figure 4.3 shows that lower-strength bonds delay the capture event, which occurs when the tumor cell has zero velocity. This result is very useful, as it implies that a given system may have a "critical bond strength" such that the capture event will not occur.

While some of the oscillations in Figure 4.3 is numerical, much of the noise is caused by the stochastic nature of the bond formation and breakage, since the cell velocity changes abruptly as bonds are created and broken. Due to these oscillations, the capture event can be defined using a moving average of the velocity magnitude.

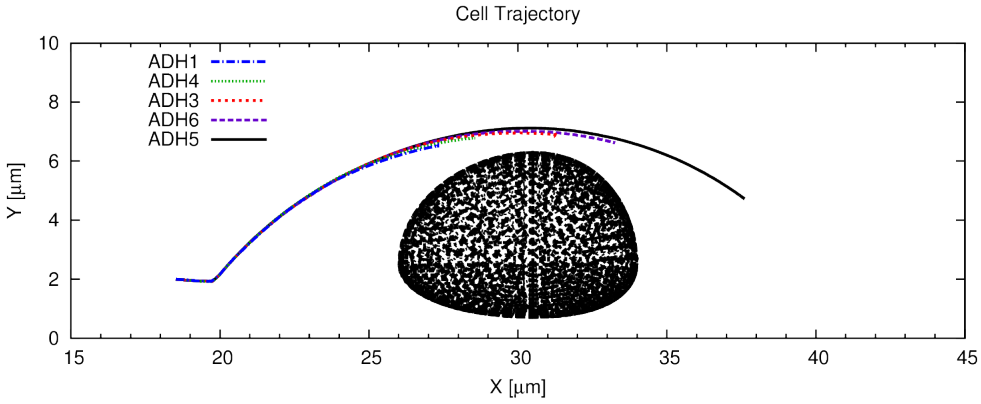

Figure 4.2: Cell trajectories for each of the cases when adjusting biochemistry parameters.

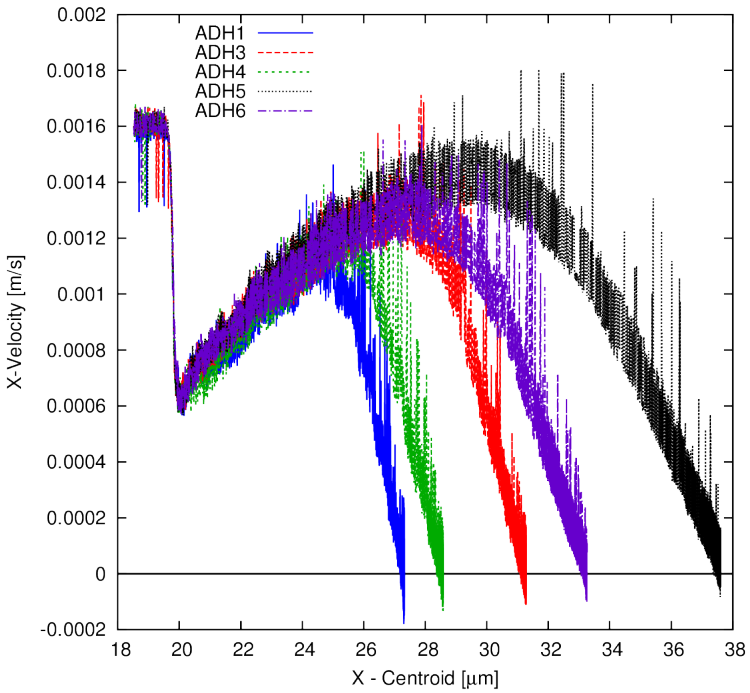

Figure 4.3: Cell velocity versus Centroid location for each of the five cases.

4.1.2.2 Effect of Flow System on Adhesion Model

Where the first set of simulations changed the parameters of the biochemistry model, this parametric study uses a fixed set of bond spring constants and varies system level parameters. By changing the shear rate of the flow and the relative position of the two bodies, it is possible to find the combination of system parameters needed to achieve a critical bond strength equivalent to that fixed in the biochemistry model.

Physically, the critical bond strength is determined by the hydrodynamic forces acting on the body. If the bond strength is not great enough to counteract the hydrodynamic forces, the bond will be extended and have an increased probability of breakage. Thus, critical bond strength is system dependent and can be altered by varying flow parameters. This phenomenon is consistent with that observed by Potanin, et al [16].

This exercise is important for several clinical applications. For example, it may be possible to prescribe a drug that inhibits aggregate formation by adjusting the critical bond strength to be greater than the known bond strength of a given tumor cell type.

To demonstrate the effects of the flow system on the adhesion process, three cases were run using the parameters in Tables 4.4 and 4.5. Each of the cases corresponds to one of the possible outcomes of the simulation. The first case contains neither bond formation nor the formation of an aggregate. The second case shows bond formation between the cells but an aggregate does not form. In the third case, there is bond formation and an aggregate is formed.

The trajectories of the three cases can be seen in Figure 4.4.

Table 4.4: Effect of Flow System on Adhesion Model - System Constants

Parameter	Value
k^0_{off} [s^{-1}]	0.3
k^0_{on} [s^{-1}]	3×10^{-3}
ϵ [μm]	1.2
λ [μm]	0.05
Domain Size [μm]:	
X	60
Y	32
Z	42
PMN Initial Centroid [μm]:	
X	30
Y	2.5
Z	21

Table 4.5: Effect of Flow System on Adhesion Model - Variable Parameters

	No Interaction	Collision w/o Adhesion	Collision w/ Adhesion
$\dot{\gamma}$ [s^{-1}]	220	113	205
$X_{\text{TC Centroid, } t_0}$	18.5	18.5	18.5
$Y_{\text{TC Centroid, } t_0}$	15.6	12.7	8.6
$Z_{\text{TC Centroid, } t_0}$	21.0	21.0	21.0

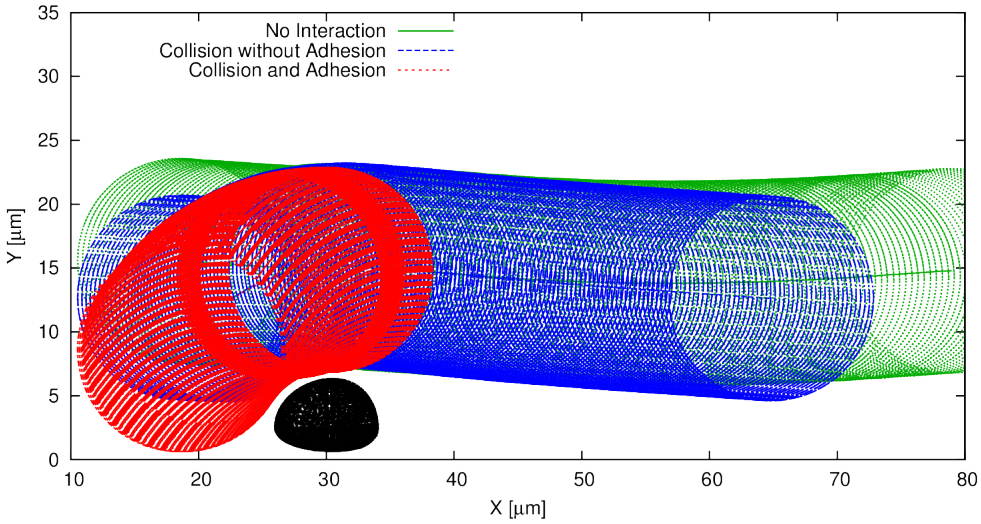

Figure 4.4: Cell trajectories for each of the three cases when adjusting system-level parameters.

4.2 Arbitrarily Many Cell Simulation

In order to simulate real blood flow systems, it is necessary to discretely model a large number of bodies. This requirement is met through use of the n-body generalization and cell-cycling approach described previously.

To demonstrate this capability, two cases are shown. The first of which, shown in Figures 4.5 and 4.6, computes full biochemistry and 6DOF motion for eight bodies; six of the bodies are RBCs, one is an adhered white blood cell, and one is a tumor cell. In this simulation the RBCs are spaced such that no collision should occur. This allows the RBCs to flow solely due the the hydrodynamics forces acting on each of the bodies. The RBCs travel through the flow much faster than the white blood cell and tumor cell because they are inherently located closer to the blood vessel centerline where the maximum flow velocity occurs, thus the RBCs may be cycled through the flow domain many times before the tumor cell has reached the exit buffer area. In figure 4.5, several of the RBCs can be seen near the exit buffer zone. After several timesteps, these RBCs will have completely entered the exit flow domain and cycled through the domain (as shown in Figure 4.6). By flowing continuously through the domain, the cells can evolve to a nearly statistically stationary solution.

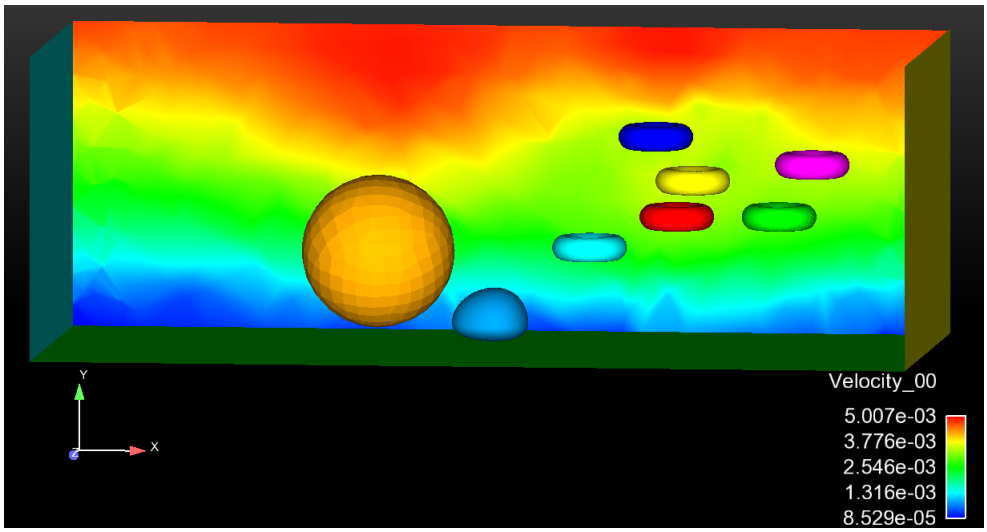

Figure 4.5: Large flow domain with eight cells at t=0. Six RBCs are placed in random locations downstream of the TC and PMN. Flow from left to right.

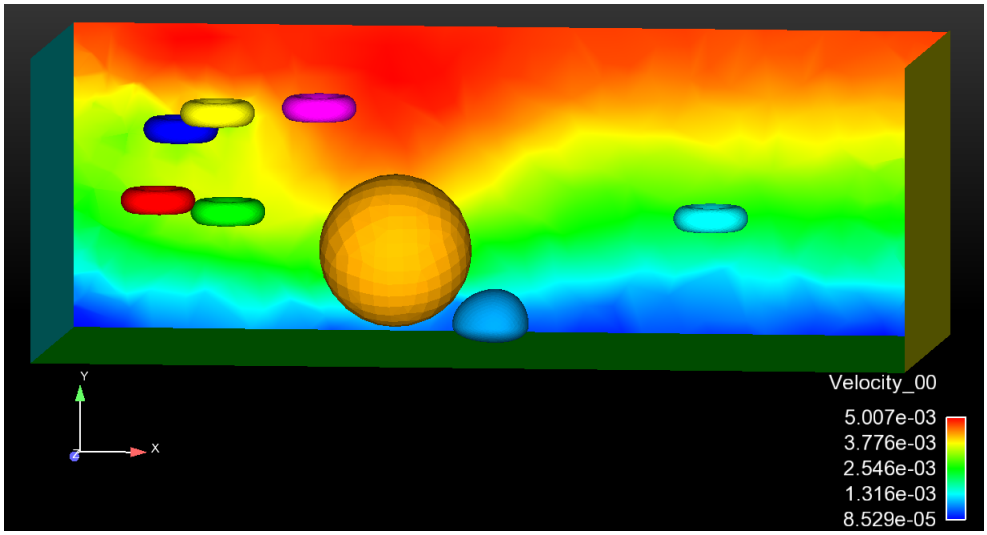

Figure 4.6: Large flow domain after several timesteps. Most of the RBCs have moved through the flow domain in accordance with the cyclic boundary condition described in Section 3.5.1. Flow from left to right.

In the second case, fifteen cells of various shapes are simulated. Twelve of the bodies are RBCs, one is an adhered white blood cell, one is a flowing white blood cell, and one is a tumor cell. These cells are much closer than those in the previous simulation, as shown in Figure 4.7, making this problem much more complex. However, the computational platform is still robust enough the handle the system.

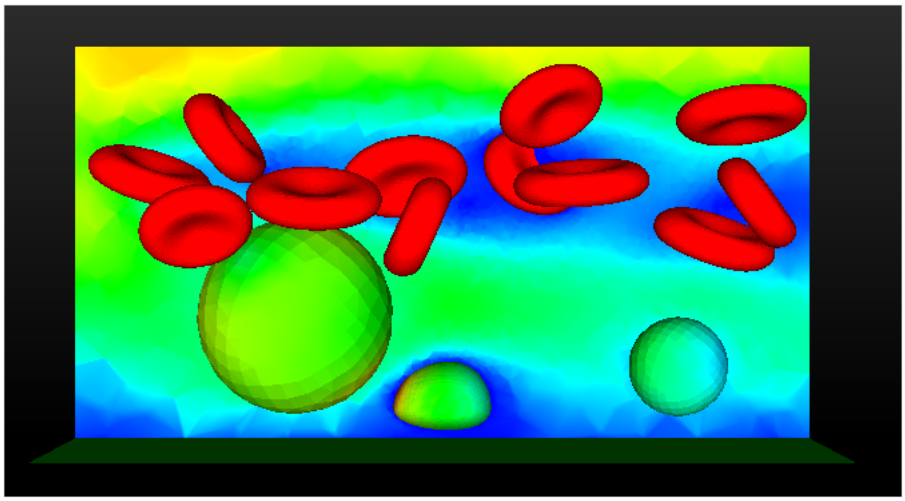

Figure 4.7: Area of interest of flow domain with fifteen cells. Three different typed of cells are present in this simulation.

Given the close proximity of all the cells, the repulsion model described in Section 2.2.3 plays a critical role in this case. All cells are able to move through the flow domain while maintaining a minimum separation distance. The simulation shown in Figure 4.7 is the first of many densely-packed cell simulations to be solved using the modeling tool developed in this study. These densely-packed simulations are important, as modeling of physiologically accurate blood flow systems involved a large number of densely-packed cells.

Chapter 5

Conclusions and Future Work

5.1 Conclusion

The major development of this thesis was the creation of a robust computational platform capable of solving multi-physics problems for biological applications. Much of this work involved generalizing the physics formulations such that they could be applied to systems of any size. In addition, several issues regarding high performance computing were addressed to increase the computational efficiency of the platform. By addressing these concerns, the tool is many times faster than the previous generation of the code.

The major contributions of this thesis include stability analysis of the coupled 6DOF-hydrodynamics system, leading to an efficient adaptive timestep specification, introduction of the immersed boundary method (IBM), wherein arbitrary cell shapes are meshed internally, to enable exact inertial property evaluations, and generalization to an n-body system including a novel cyclic method for reintroducing cells on periodic boundaries to enable statistical stationarity. Other contributions of this thesis include an approach to couple various physics with the 6DOF-hydrodynamics system, a fully general and parallelized algorithm for the modeling of biochemical interactions, and an overall reduction in the run time of for these simulations.

Given the speedup of the code and robustness of the generalized physics formulations, the tool has the capability to be used in a clinical setting. This work is ready to be applied to future generations of the overall, multiscale simulation

effort, wherein the cell-molecular scale is interfaced with population scale modeling to predict clinical environment aggregation behavior.

5.2 Future Work

While this work has accomplished the goal of developing a robust computational platform, there are still many improvements to be made.

The first improvement is to reintroduce the structural mechanics modeling [11, 12]. In the spirit of computational efficiency, the structural mechanics should be performed within the flow solver. Such an implementation would allow for better coupling of the fluid-structure interactions through use of the immersed boundary approach. By implementing a finite-volume approach, the structural mechanics may be solved simultaneously with the fluid mechanics.

Once the structural mechanics has been implemented, the structural mechanics and 6DOF motion calculations should be implicitly coupled. Handling the numerics of these physics implicitly would remove the temporally stability restriction imposed by the explicit Euler method. Therefore, the time step would be determined solely by the necessary temporal resolution.

Future generations of this code should be used to solve clinical problems in a timely fashion. Unfortunately, limitations of computing resources makes some problems intractable. To approach these types of problems, it is possible to perform multiscale modeling through use of statistical emulation (e.g., reduced-order modeling). By running many simulations with a wide range of system parameters, it is possible to statistically emulate the computational tool. This emulation is a black-box mathematical model trained using empirical data capable of producing fast predictive models. To generate a fast predictive model for a given blood flow system, statistical emulators based on a generalized Gaussian Process approach may be used and trained using data obtained from the aforementioned tool. This emulator, used in conjunction with the tool, may give the performance and accuracy needed to solve clinical problems.

Bibliography

[1] Epstein, F. H. and H. F. Bunn (1997) "Pathogenesis and treatment of sickle cell disease," *New England Journal of Medicine*, **337**(11), pp. 762–769.

[2] Ingram, V. M. (1957) "Gene mutations in human haemoglobin: the chemical difference between normal and sickle cell haemoglobin," *Nature*, **180**(4581), pp. 326–328.

[3] Pauling, L., H. A. Itano, S. Singer, and I. C. Wells (2004) "Sickle Cell Anemia, a Molecular Disease," *Landmarks in Medical Genetics: Classic Papers with Commentaries*, **51**, p. 200.

[4] Platt, O. S., D. J. Brambilla, W. F. Rosse, P. F. Milner, O. Castro, M. H. Steinberg, and P. P. Klug (1994) "Mortality in sickle cell disease–life expectancy and risk factors for early death," *New England Journal of Medicine*, **330**(23), pp. 1639–1644.

[5] (2014), "Red Blood Cell flow: Normal versus Sickle-Cell," Electronic Photo. URL `http://www.nhlbi.nih.gov/health/health-topics/topics/sca/`

[6] Burt, V. L., P. Whelton, E. J. Roccella, C. Brown, J. A. Cutler, M. Higgins, M. J. Horan, and D. Labarthe (1995) "Prevalence of hypertension in the US adult population results from the Third National Health and Nutrition Examination Survey, 1988-1991," *Hypertension*, **25**(3), pp. 305–313.

[7] Chaouat, A., E. Weitzenblum, and T. Higenbottam (1996) "The role of thrombosis in severe pulmonary hypertension," *European Respiratory Journal*, **9**(2), pp. 356–363.

[8] Fedullo, P. F., W. R. Auger, K. M. Kerr, and L. J. Rubin (2001) "Chronic thromboembolic pulmonary hypertension," *New England Journal of Medicine*, **345**(20), pp. 1465–1472.

[9] Sowers, J. R., M. Epstein, and E. D. Frohlich (2001) "Diabetes, hypertension, and cardiovascular disease an update," *Hypertension*, **37**(4), pp. 1053–1059.

[10] (2014), "Effects of Hypertension on Blood Vessel," Electronic Photo. URL `http://kidney.niddk.nih.gov/kudiseases/pubs/highblood/`

[11] Hoskins, M. (2008) *Development of a computational fluid dynamics tool to explore the interactions between cancer cells and leukocytes*, Ph.D. thesis, The Pennsylvania State University.

[12] Hoskins, M., R. Kunz, J. Bistline, and C. Dong (2009) "Coupled flow–structure–biochemistry simulations of dynamic systems of blood cells using an adaptive surface tracking method," *Journal of fluids and structures*, **25**(5), pp. 936–953.

[13] Behr, J. (2013) *Multiscale modeling of cancer cell adhesion*, Master's thesis, The Pennsylvania State University.

[14] Bell, G. et al. (1978) "Models for the specific adhesion of cells to cells," *Science*, **200**(4342), pp. 618–627.

[15] Dembo, M., D. Torney, K. Saxman, and D. Hammer (1988) "The reaction-limited kinetics of membrane-to-surface adhesion and detachment," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, **234**(1274), pp. 55–83.

[16] Potanin, A., V. Verkhusha, and P. Vrzheshch (1993) "Coagulation of particles in shear flow: applications to biological cells," *Journal of colloid and interface science*, **160**, pp. 405–405.

[17] Hammer, D. and S. Apte (1992) "Simulation of cell rolling and adhesion on surfaces in shear flow: general results and analysis of selectin-mediated neutrophil adhesion," *Biophysical journal*, **63**(1), pp. 35–57.

[18] Migliorini, C., Y. Qian, H. Chen, E. Brown, R. Jain, and L. Munn (2002) "Red blood cells augment leukocyte rolling in a virtual blood vessel," *Biophysical journal*, **83**(4), pp. 1834–1841.

[19] Jadhav, S., C. Eggleton, and K. Konstantopoulos (2005) "A 3-D computational model predicts that cell deformation affects selectin-mediated leukocyte rolling," *Biophysical journal*, **88**(1), pp. 96–104.

[20] Kundu, P. and I. Cohen (2008) *Fluid Mechanics*, Elsevier Academic Press.

[21] Behr, J., B. Gaskin, C. Fu, C. Dong, and R. Kunz "Localized Modeling of Biochemical and Flow Interactions During Cancer Cell Adhesion," .

[22] Ionescu, H. (2010), "Six degrees-of-freedom," Wikimedia Commons.

[23] Park, E. Y., M. J. Smith, E. S. Stropp, K. R. Snapp, J. A. DiVietro, W. F. Walker, D. W. Schmidtke, S. L. Diamond, and M. B. Lawrence (2002) "Comparison of PSGL-1 Microbead and Neutrophil Rolling: Microvillus Elongation Stabilizes P-Selectin Bond Clusters," *Biophysical Journal*, **82**(4), pp. 1835 – 1847.

[24] Burkett, R. D. (2006), "Digestive System Lab," .
URL `http://faculty.southwest.tn.edu/rburkett/Digest34.jpg`

[25] Van Doormaal, J. and G. Raithby (1984) "Enhancements of the SIMPLE method for predicting incompressible fluid flows," *Numerical heat transfer*, **7**(2), pp. 147–163.

[26] Peskin, C. S. (2002) "The immersed boundary method," *Acta numerica*, **11**(0), pp. 479–517.

[27] Rasmussen, R. V. and M. A. Trick (2008) "Round robin scheduling–a survey," *European Journal of Operational Research*, **188**(3), pp. 617–636.