



All Theses and Dissertations

2012-04-21

Multi-Processor Computation of Thrombus Growth and Embolization in a Model of Blood-Biomaterial Interaction Based on Fluid Dynamics

Brandon Thomas Andersen
Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>

 Part of the [Chemical Engineering Commons](#)

BYU ScholarsArchive Citation

Andersen, Brandon Thomas, "Multi-Processor Computation of Thrombus Growth and Embolization in a Model of Blood-Biomaterial Interaction Based on Fluid Dynamics" (2012). *All Theses and Dissertations*. 3465.
<https://scholarsarchive.byu.edu/etd/3465>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Multi-Processor Computation of Thrombus Growth and Embolization

in a Model of Blood-Biomaterial Interaction

Based on Fluid Dynamics

Brandon Andersen

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Kenneth A. Solen, Chair
S. Sukavaneshvar
Brad C. Bundy

Department of Chemical Engineering

Brigham Young University

June 2012

Copyright © 2012 Brandon Andersen

All Rights Reserved

ABSTRACT

Multi-Processor Computation of Thrombus Growth and Embolization in a Model of Blood-Biomaterial Interaction Based on Fluid Dynamics

Brandon Andersen
Department of Chemical Engineering, BYU
Master of Science

This work describes the development and testing of a real-time three-dimensional computational fluid dynamics simulation of thrombosis and embolization to be used in the design of blood-contacting devices. Features of the model include the adhesion and aggregation of blood platelets on device material surfaces, shear and chemical activation of blood platelets, and embolization of platelet aggregates due to shear forces. As thrombus develops, blood is diverted from its regular flow field. If shear forces on a thrombus are sufficient to overcome the strength of adhesion, the thrombus is dislodged from the wall. Development of the model included preparing thrombosis and embolization routines to run in a parallel processing configuration, and estimating necessary parameters for the model including the adhesion strength of platelet conglomerations to the device surfaces and the criterion threshold for the coalescence of neighboring thrombi. Validation of the model shows that the effect of variations in geometry may be accurately predicted through computational simulation. This work is based on previous work by Paul Goodman, Daniel Lattin, Jeff Ashton, and Denzel Frost.

Keywords: Brandon Andersen, thrombosis, embolization, computational fluid dynamics, simulation, parallel processing, shear and chemical mediated platelet activation

TABLE OF CONTENTS

TABLE OF CONTENTS	v
LIST OF TABLES	vii
LIST OF FIGURES	ix
1 Introduction.....	1
2 Background	3
2.1 Thrombosis and Embolization	3
2.2 Computational Models of Thrombosis and Embolization	4
2.2.1 Platelet Adhesion	4
2.2.2 Protein Adsorption	5
2.2.3 Comprehensive Models	5
2.2.4 Computational Fluid Dynamics Models	7
2.2.5 Other Models	11
2.3 Experimental Studies of Thrombosis	12
3 Objectives.....	15
4 Constructing the Model	17
4.1 Parallelize the Thrombosis and Embolization Code.....	19
4.2 Simulate the Flow of Blood Around Thrombi.....	22
4.3 Perform Detailed Tracking of Targeted Cells and Thrombi.....	24
4.4 Optimize the Code to Minimize Runtime	25
4.5 Estimate Necessary Model Parameters.....	29
5 Predicting the Effect of Insert Spacing In the Flow Cell.....	39
6 Demonstrate the Model’s Use for Designing a Hemodialysis Catheter	47

7 Conclusion and Recommendations	53
REFERENCES.....	57
APPENDIX A FORTRAN ROUTINES	63
A.1 Sorsca.f	63
A.2 Posdat.f.....	70

LIST OF TABLES

Table 4-1 Processes in computational cells and defined thrombus	17
Table 4-2 Summary of Model Improvements.....	18
Table 4-3 Central composite design with tabulated results	32
Table 4-4 Regression coefficients for number of emboli as a function of adhesion strength and coalescence threshold.....	32
Table 4-5 Results from flow cell experiments.....	35
Table 4-6 Estimated values for thrombosis model	37
Table 5-1 Summary of insert spacing computational study results	41
Table 5-2 Experimental data from insert spacing comparison	43
Table 5-3 Effect of insert spacing on average number of emboli (averaged from the data in Table 5-2).....	44

LIST OF FIGURES

Figure 2-1 Experimental flow cell	13
Figure 4-1 Flow cell geometry showing typical processor-domain boundaries for 64 processors	19
Figure 4-2 Flowchart representing thrombosis and embolization routine algorithm with (right) and without (left) MPI implementation for parallel processing.....	21
Figure 4-3 Example of successful growth of thrombi (shown in red) across processor domain boundaries	21
Figure 4-4 (a) Predicted thrombosis and (b) velocity profile (m/s) from the Lattin-Ashton-Frost model.....	23
Figure 4-5 Velocity profile (m/s) (a) before and (b) after thrombus growth (thrombus indicated by white region in figure b).....	24
Figure 4-6 Coalescence and embolization routine (a) before and (b) after optimization	25
Figure 4-7 Cell stencil from Lattin-Ashton-Frost model.....	26
Figure 4-8 Average embolus volume vs. cell density.....	27
Figure 4-9 Thrombus location map.....	27
Figure 4-10 Established standard cell stencil (2.5×10^{15} cells/m ³).....	28
Figure 4-11 Average embolus volume vs. residual tolerance.....	29
Figure 4-12 Central composite design with additional points for curvature test.....	31
Figure 4-13 Response surface for number of emboli.....	33
Figure 4-14 Contour plot for number of emboli	33
Figure 4-15 Number of emboli data with 95% confidence interval	36
Figure 4-16 Optimization plot for adhesion strength and coalescence threshold.....	36

Figure 5-1 Variations on flow cell to demonstrate effect of device design	39
Figure 5-2 Effect of insert spacing on velocity field	40
Figure 5-3 Fraction of time occupied with thrombus	42
Figure 5-4 Main effect of all independent variables in spacing experiments	43
Figure 5-5 Paired t-test of average number of emboli as effected by insert spacing.....	45
Figure 6-1 Standard Mahurkar blood dialysis catheter design	47
Figure 6-2 Thrombus history in standard Mahurkar catheter after 330 seconds	48
Figure 6-3 Velocity vector field in standard Mahurkar after 330 seconds	49
Figure 6-4 Augmented Mahurkar blood dialysis catheter design.....	50
Figure 6-5 Thrombus history in augmented Mahurkar catheter after 330 seconds	50
Figure 6-6 Velocity vector field in augmented Mahurkar after 330 seconds	51

1 INTRODUCTION

Since the 1980's, blood-contacting devices, including artificial hearts, catheters, grafts, heart valves, and stents, have made up a large portion of the medical device industry[1]. These devices are largely responsible for dramatic improvements in cardiovascular-related mortality including, for example, a 40% decrease in heart attack mortality from 1980 to 2000[2]. The benefits of blood-contacting devices are clear, however their use is not without risk. Thrombosis, the growth of blood aggregates on biomaterial surfaces, and thromboembolism, the subsequent embolization of those blood aggregates, have been documented in blood-contacting devices as early as 1959[3-5] and are a major cause of death for cardiovascular device patients. Despite more than a half century of intense research, the principles of the physical and chemical design of materials and devices that control the blood-material interactions which result in thromboembolic events are not fully understood[6].

“Virchow’s Triad” identified blood, surface, and flow as the three factors that govern thrombosis and embolization over 100 years ago[7]. Since that time, and especially in the last 30 years, since the advent of the cardiovascular stent, researchers have sought to better understand the mechanisms of thrombosis. Research has identified many key blood chemicals and components and their role in thrombosis; comparatively “bioinert” materials, appropriate for use in the body, have been identified through experimentation; and the advances in Computational Fluid Dynamics (CFD) have enabled a much more thorough understanding of the flow of blood through and around devices. However, because of the complexity of, and interaction between,

these factors[8, 9], the development of a comprehensive three-dimensional transient computational model of the formation, growth, and embolization of thrombi associated with cardiovascular devices has proven difficult. Such a computational model would be invaluable in the successful design of devices that minimize the threat of thromboembolic events.

2 BACKGROUND

2.1 Thrombosis and Embolization

The events of thrombogenesis can be grouped into the following three categories: 1) protein activation and adsorption, 2) platelet and leukocyte activation and attachment, and 3) thrombus growth and detachment[10, 11]. This categorization of events permits the systematic examination of the processes known to contribute to thrombosis and thromboembolism.

All materials exposed to blood adsorb within seconds a thin layer of proteins[12], mostly fibrinogen[6]. Multilayers of protein continue to build on the initial monolayer of proteins. In the absence of anticoagulants, contact with the surface causes proteins to adhere, and in the case of complement and coagulation proteins, activates the proteins and initiates a cascade of reactions that result in the creation of thrombin and fibrin, which in turn activate more proteins, and blood platelets, forming a mesh for the conglomeration of those proteins and platelets[11].

Chemical agonists, such as adenosine diphosphate (ADP)[13], and shear forces[14] activate platelets in the blood stream and previously unactivated adherent platelets. Activated platelets become sticky[15], deform to irregular spheres with spiny pseudopods[16], and release agonists[13], which activate neighboring platelets. The activated platelets (as well as unactivated platelets) carried to the wall by diffusion and convection adhere easily to protein molecules and other activated platelets on a material surface[17]. Leukocytes, or white blood cells, and complement proteins are believed to contribute to thrombosis[11], however their relative importance in the mechanisms of thrombosis and thromboembolism is not known[18].

Thrombosis proceeds through the reaction cascade initiated by the activated proteins, and relies on the continued transport of proteins and platelets to the surface of the forming thrombus. As the thrombus grows, it disrupts the flow patterns of the surrounding blood, which causes increased flux of proteins and platelets to the surface, as well as increased shear rates. As the thrombus continues to grow into the flow path, the shear stress may overcome the adhesion strength of the thrombus to the surface, causing it to break off of the material surface and enter the flow stream[19]. Over time, the fibrin may break down or the monolayer proteins may be denatured, also resulting in a release from the wall[10, 20].

2.2 Computational Models of Thrombosis and Embolization

Many models have been developed which attempt to describe thrombosis and/or thromboembolization. While most aspects of the processes have been modeled in some way, most models focus on a single component rather than the process as a whole, and are devised in such a way that precludes their extension to transient flow through a three-dimensional geometry.

2.2.1 Platelet Adhesion

In 1972 Peter D. Richardson[21] was the first person, known to the author, to model thrombosis. His model, which focused only on platelet activation by chemical agonists, consisted of platelets in steady concentration flowing in a linear velocity profile next to a wall. A maximum thrombus radius was assumed and used to define a “concentration layer” of agonist around a pre-existent seed thrombus. The agonist concentration layer was assumed sufficient to activate any platelet that entered it after a characteristic time. If an activated platelet remained in the immediate vicinity of the thrombus (defined by the agonist concentration layer) when

activation time was reached, it was added to the thrombus volume. The model, although primitive, was consistent with the data collected two years earlier by Begent and Born[22], which suggested that the growth rate was exponential in time and was zero above some critical velocity. Richardson called his work, “a simplified model that predicts observed behavior.”[21] In 1977, Ruckenstein et al.[23] added a kinetic expression for the adhesion of platelets to the thrombus mass which produced a plateau in thrombus growth rate, consistent with experimental observation[24].

2.2.2 Protein Adsorption

Many models of protein adsorption exist[25-30]. One of the most detailed of these models was developed by Madrusov et al.[28]. It predicts in real time the competitive and displacement adsorption of specific proteins. Because protein adsorption is a complicated process, and only a small part of a comprehensive thrombosis model, the reader is referred to the review of significant protein adsorption models by Madrusov et al.[28] for additional information on this topic.

2.2.3 Comprehensive Models

Marmur and Cooper[31] combined protein adsorption and detachment to platelet activation and deposition in their 1982 model, creating the first computational model that incorporated protein and platelet adhesion as well as thrombus growth and embolization. The model broke the process of thromboembolization into the following steps: (1) deposition of a first layer of platelets directly on the bare surface, (2) multilayer deposition of platelets on the first platelet layer, (3) adsorption of a first layer of protein directly on the bare surface, (4) multilayer adsorption of protein onto the basal protein layer, (5) Deposition of a first layer of

platelets on a protein layer, (6) multilayer deposition of platelets on the first layer of platelets deposited on a protein layer, and (7) protein desorption after a characteristic time representative of protein denaturation (proteins desorbed were assumed to carry with them all of their protein and platelet deposits). The expression for each for these steps included a kinetic parameter that was fixed in order to be consistent with experimental observation of growth rate over time. Equations were solved using the 4th-order Runge-Kutta method with an initial condition of no protein or platelet surface coverage. Activation of proteins and platelets was not treated. This model was later extended by Wilson et al.[32] in 1986 to include an eighth step: (8) embolization of thrombi due to hemodynamic forces. The underlying principle of the mechanism was that thrombi that attained a critical size were subject to forces that removed them from the surface. This removal carried away everything deposited beneath the thrombi.

Marmur and Cooper's model, with Wilson et al.'s extension, was important in that it was the first comprehensive model of thromboembolism; however it did not predict thromboembolism in a way that could be applied to actual cardiovascular devices[33]. Surface coverage of proteins and platelets was a controlling variable in the kinetic expressions used, as would be expected in an adsorption model, however this variable was also used to define the characteristic size and height of a thrombus by taking an average height of "layers" of surface coverage. Because the size and shape of the thrombus was not well defined, an account of its effect on local hemodynamics was not possible, nor was embolization due to realistic hemodynamic forces.

Reynolds et al.[10] improved on the Wilson et al. model in 1993 by considering the coexistence of three distinct proteins: platelet-adherent proteins (fibrinogen), 'other proteins' (albumin), and active coagulation factors (Factor XII). The proteins functioned separately in that

platelets were allowed to deposit only onto ‘fibrinogen’ proteins, and Factor XII participated in the thrombosis processes by producing fibrin and fibrin-related platelet-platelet bridges.

2.2.4 Computational Fluid Dynamics Models

In more recent years, Computational Fluid Dynamics (CFD) has become a central component of many of the models developed to simulate thrombosis. Many groups are using CFD in three-dimensional geometries at steady state in order to investigate shear stress and stagnation time, which are good predictors of thrombosis and embolization[34-40].

Unfortunately these models only give a probability map of where thrombi are most likely to form in the device without providing any kind of a prediction of their size, growth rate, or embolization.

In several studies, models of protein and/or platelet functional behavior in addition to two-dimensional CFD were used to predict thrombosis in various geometries[7, 33, 41]. Sorensen et al.[7] developed such a model incorporating platelet activation, agonist synthesis and release by activated platelets, and thrombin generation and adhesion. The model required the estimation of four parameters to fit it to experimental data: shear-dependent platelet diffusivity and three platelet-deposition-related reaction rate constants. After determining values for the model parameters for platelet deposition onto a collagen substrate by fitting to experimental data, the model was validated by comparison to literature data for flow between parallel plates. The model provided good predictions of deposition trends and magnitudes for data taken in the presence of heparin, but not in the presence of citrate. The model, which was never extended to three dimensions, also failed to include embolization of platelet aggregates. Goodman et al.[33] developed a similar model, borrowing heavily from the reaction expressions and constants compiled by Sorensen et al., but included both single platelet and platelet aggregate embolization

and alterations of local fluid dynamics due to the presence of a thrombus. His model was useful in the prediction of initial or acute thrombosis, but was limited in the time duration of simulations because it predicted that thrombus eventually formed on all surfaces in the model. Daniel Lattin, Jeff Ashton, and Denzil Frost performed subsequent work to port Goodman's model to a new commercial CFD package, STAR-CD® (CD-adapco, Melville, NY) which allowed easier modeling of more complex medical devices by its auto-meshing tool and extension of Goodman's model to three dimensions, and improved the prediction of thromboembolism due to shear forces. While they were successful in developing a working model, several aspects of the model were not validated, and preliminary results suggested that the code was not completely working as intended. Areas of concern in their model include the following:

Parallel processing. The Goodman and Lattin-Ashton-Frost models were written without attention to the complications of multi-processor parallel computing, a feature of supercomputers that significantly reduces computation time. STAR-CD® uses the MIMD (Multiple Instruction stream, Multiple Data stream) technique for parallel processing, which is the most common parallel arrangement. In MIMD processing, one code is written and each processor runs the code on independent data. In the case of an MIMD problem utilizing domain decomposition, each processor runs the code for the fraction (domain) of the model assigned to that processor. Domain decomposition is performed on the three-dimensional model space automatically by STAR-CD®. Each requested processor is assigned an approximately equally sized subsection of the total volume for which it solves the transport equations necessary to give the flow solution and runs code predicting thrombosis and embolization. Also, the hardware of the BYU supercomputer is set up to use distributed memory (each processor has its own memory), which

is highly scalable but is also more difficult to program than a shared memory arrangement (in which every processor has direct access to one global memory space). Complications arise with the domain decomposition in our problem where we are not only solving for micro-scale 3-dimensional flow, but also describing macro-scale growth of thrombi. Some thrombi will initiate in one processor domain but will extend into the domain of another processor, which may require that one processor has access to memory stored on other processors. Simulations by the Lattin-Ashton-Frost model where such extension was not accommodated predicted the formation of thrombi that were bounded by domain boundaries.

Volume discretization. The method for solving flow fields using computational fluid dynamics involves discretizing the physical domain into 3-dimensional cells. The properties of the fluid within each cell are assumed to be uniform, and the equation of motion is solved in agreement with the equation of continuity in each cell. The justification of uniform conditions in a cell requires that the cell be small enough to not encompass a large change in physical properties, or velocity. The cells must also be small enough to estimate the derivative across a cell with sufficient accuracy. Examination of the volume discretization used in the Lattin-Ashton-Frost model reveals cells that are too large to ensure accuracy in these assumptions.

Flow around growing thrombi. Goodman's model employed with success a method of altering flow patterns to accommodate the growth of thrombus by increasing the viscosity of cells designated as thrombus. Viscosity was increased from $.0035 \text{ Pa} \cdot \text{s}$ to $1 \text{ Pa} \cdot \text{s}$ to stop fluid flow within the cell and cause the surrounding fluid to flow around the thrombus cell. The alteration of the flow field by the presence of thrombus is important to the simulation in the calculation of growth rates and the shear force exerted on the thrombus, which may cause it to embolize.

Lattin, Ashton, and Frost's version of the model in STAR-CD® attempted to employ the same method, however it has been found that for computational cells for which the viscosity was reported to increase as desired, the flow field remained unaltered.

Input parameters. The predictions of the model are based upon a system of equations involving 1) the rate of transport and adhesion of both resting and active platelets to the device wall and existing thrombus, 2) the convective and diffusive transport and production and consumption of chemicals involved in the activation of platelets, and 3) the activation of platelets due to both shear forces and chemical agonists. While all of these processes are included in the model presented by Goodman and later by Lattin, Ashton, and Frost, large discrepancies exist between model predictions and observed behavior. The model predicts thrombosis in locations and formations that differ from experimental observations, and the predicted rate of thrombosis is many times too fast. These results suggest that a careful critical review of calculations and input parameters should be performed.

In addition to a critical review of the computational code governing the processes of thrombosis and embolization, two key parameters must be estimated. The model requires values for the strength of adhesion of a mass of platelets to the wall of a blood-contacting device (measured in N/m^2) and requires a criterion for the coalescence of neighboring thrombi. Experimental observation suggests that as two thrombi grow in the same region, they may become infused to the point that they will no longer embolize and separate thrombi, but as a single mass. In our code the criteria for this coalescence effect is that a specific threshold percent of total surface area of a thrombus be in contact with an adjoining thrombus. The estimation of these two parameters is essential to achieving accurate results from the simulation.

Runtime efficiency. Lattin, Ashton, and Frost made significant progress toward developing a code that described the growth and embolization of thrombi, but were not able to address the issue of run time efficiency. Ten minutes of simulation time using their model in STAR-CD® typically required up to a week running on the BYU supercomputer, but that time likely could be reduced by judicious optimization of the code. Areas of concern have been identified to improve the runtime efficiency. These include 1) removing redundant and unnecessary calculations and data storage, 2) using a volume mesh density that assures grid independence without requiring excessive runtime, and 3) setting the residual tolerance for the convergence process to achieve adequate precision without undue iteration.

Calculation tracking. An obstacle to finding errors in the model is that the standard methods of presenting results in Star-CD are limited to a series of plots of the transient solution of variables such as thrombus presence and fluid velocity. Running the model on the supercomputer (which is necessary at this point) does not allow a user to step through the calculations and verify that the model is functioning as designed. Tools need to be developed within the user code to follow all the calculations and progression of thrombosis in a single cell or of a single thrombus in the model to help identify the causes of existing errors and their solutions and ultimately to increase confidence in the model's predictions. These same tools will allow the user to collect data on thrombus growth and embolism events which STAR-CD® does not natively provide.

2.2.5 Other Models

Aaron Fogelson has been building a model of platelet interaction with a damaged vessel wall in connection with several researchers including Robert Guy[42] and K. Leiderman[43].

These models take the unique approach of tracking every individual platelet. While computationally expensive, Fogelson's model incorporates the activation of platelets, adhesion to a damaged endothelial wall and to other activated platelets, thrombus growth and effect on flow patterns, and embolization by shear forces all in real time in 2-dimensional space. The model provides good prediction of thrombosis growth and embolization trends. Predicting thrombosis locations in a medical device with Fogelson's model would require expansion to three dimensions and much larger systems and altering parameters to reflect the difference between a damaged endothelial cell wall and an artificial material surface, which would make the computational solution far too costly for practical use.

Other methods used in the modeling of platelet thrombosis include a multi-scale model by Xu et al.[44] and a particle dynamics model by Filipovic et al.[45]. These models focus on venous thrombus formation through the intrinsic reaction cascade, and arterial thrombus formation through platelet activation and adhesion, respectively. Neither model endeavors to predict thrombosis and embolization occurring in real time.

2.3 Experimental Studies of Thrombosis

Any computational model to be used in predicting thrombosis and embolization must be validated by comparison to experimental observation. Goodman's model was no exception. In connection with his computational model, Goodman designed and performed experiments to observe the processes of thrombosis and thromboembolization[33]:

Human blood was drawn into heparin (final concentration = $1 U/ml$) from donors free of platelet-affecting drugs and was perfused through cylindrical flow cells at $0.75 ml/min$. The flow cells (see Figure 2-1) were constructed from $580\text{-}\mu m$ ID Polyethylene (PE) tubing into

which smaller-diameter tubing segments (also PE, 280- μm ID) had been inserted to produce abrupt contractions and expansions in the flow path.

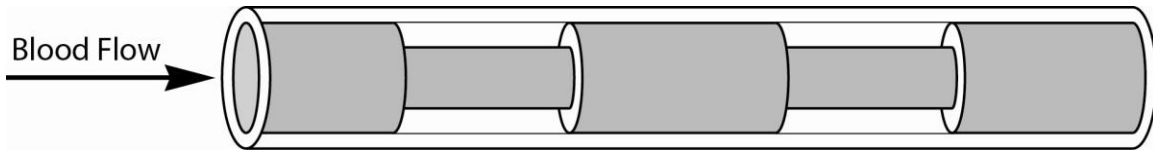


Figure 2-1 Experimental flow cell

Thrombus initiation, growth, and embolization were observed with videomicroscopy, while embolization was confirmed by light scattering, and platelet adhesion was determined by scanning electron microscopy. Thrombus growth rate was recorded as the change in thrombus height (distance from surface to furthest point of thrombus) over time. Thrombus was observed in three key areas: Area 1) the sudden contraction, Area 2) the sudden expansion, and Area 3) the reconnection point after the sudden expansion. The constricted flow section contained a lack of adhered platelets.

Whereas most thrombosis experiments focus on the state of thrombus after a set growth period, Goodman's model is unique in providing real-time monitoring of both thrombosis and embolization. This makes it easily comparable to the results of a computational model, which also provides real-time growth and embolization patterns.

Corbett et al.[34] performed similar experiments involving blood flow through a step change. Corbett's experimental setup consisted of bovine blood heparinized at a concentration of 3 U/ml heparin in a flow loop. Thrombus size and location were recorded after 2 hours of flow. Data on growth rate and embolization was not recorded. A digital replica of the experiment was used to compare the thrombus sites to wall shear rates. It was determined that thrombus initiates

at wall shear rates below 54 s^{-1} , differing from the threshold of $10\text{-}20 \text{ s}^{-1}$ as reported in Corbett et al.'s literature search [46-49]. Corbett concluded that the discrepancy might be due to the altered flow field produced by the presence of thrombi.

3 OBJECTIVES

Goodman, Lattin, Ashton, and Frost put much of the framework in place for an accurate model of blood material thrombosis and embolization; however, errors and inconsistencies prevented the model from being a reliable tool in the design of cardiovascular devices. The goal of this work was to create, from Goodman, Lattin, Ashton, and Frost's previous work, a reliable predictive model of thrombosis and embolization to be used as a tool in cardiovascular device design. This was accomplished by completion of the following objectives:

1. Constructing the Model
 - a. Parallelize the thrombosis and embolization code.
 - b. Correct the code to successfully simulate the flow of blood around each thrombus.
 - c. Perform detailed tracking of targeted cells to verify the correct functioning of the code.
 - d. Optimize the code to minimize runtime by 1) removing redundant and unnecessary calculations and data storage, 2) determining an optimum mesh density, and 3) determining an optimum residual tolerance for the convergence process.
 - e. Select a magnitude of the contact area criterion for coalescence of neighboring thrombi, and select a magnitude of the platelet adhesion strength.

2. Demonstrate the model's ability to compare flow geometries by predicting the effect of insert spacing in the flow cell.

The present model is intended to independently predict the effects of device design on thrombosis and embolization. To demonstrate the predictive nature of the model, variations on Goodman's original flow cell will be constructed with a range of insert spacing (see Figure 5-1). The effect of the variations will be studied in both the experimental setup and the computational model.

3. Demonstrate the model's use for designing a hemodialysis catheter.

Thrombosis and embolization will be simulated in a model of a standard Mahurkar blood catheter and in a Mahurkar blood catheter that has been modified in a way that is expected to decrease thrombosis and embolization. The Mahurkar blood catheter standard design includes eyelet perforations where entering blood is directed perpendicular to the catheter wall and the direction of bulk flow. It is believed that altering the angle of flow will decrease shear activation and thrombosis in the catheter.

The completed model can potentially be a powerful tool in the development of blood-contacting devices that limit the risk of thrombosis and thromboembolization. Model predictions include total thrombus mass over time, thrombus growth locations and relative rates, and embolization size and frequency. Common modeling software packages like AutoCAD® (Autodesk, San Rafael, CA) and SolidWorks® (Dassault Systèmes SolidWorks Corporation, Waltham, MA) can produce model geometries which may then be used by STAR-CD® in conjunction with the specialized thrombosis and embolization code described in this work to evaluate any device design and its risk of thromboembolic events.

4 CONSTRUCTING THE MODEL

The Lattin-Ashton-Frost model is a computational fluid dynamics simulation performed by STAR-CD®'s Pro-STAR solver using the SIMPLE algorithm for transient simulation. Transport and conservation equations are solved for unactivated platelets, activated platelets, adenosine diphosphate (ADP), thromboxane A₂ (TxA₂), prothrombin (pT), thrombin (T), and anithrombin III (aT). Source terms include boundary reactions, chemical synthesis and consumption reactions, and platelet activation due to both shear effects and chemical activation.

Table 4-1 Processes in computational cells and defined thrombus

Level	Order	Computational Processes
Cell	1	Platelets (resting and active) transport and adhere to the device surface or existing thrombus as governed by rate equations
	2	Incoming platelets cause the fraction of the cell's volume occupied by thrombus (ThrFr) to increase over time
	3	When ThrFr reaches 1.0, the cell is defined as a thrombus cell and fluid is diverted around it
	4	Neighboring cells are identified and allowed to continue the growth of the thrombus
Thrombus	5	The thrombus is evaluated against the coalescence criteria
	6	The thrombus is evaluated against the embolization criteria

The simulation functions on two levels: 1) the computational cell, and 2) the thrombus.

The processes that take place in each category are tabulated in Table 4-1.

While the Lattin-Ashton-Frost model provided a good foundation for this work, there were many improvements that needed to be made in order to prepare the simulation for use in the design of blood-contacting devices. These improvements included parallelizing the thrombosis and embolization code, directing flow around growing thrombi, implementing calculation tracking, cleaning up thrombosis and embolization algorithms, and estimating necessary parameters. These improvements are summarized in Table 4-2.

Table 4-2 Summary of Model Improvements

Previous	Improvement	Result
User subroutines written for a single processor	Redesigned and implemented subroutine algorithms for parallel processing	Allows for growth of thrombi across processor boundaries and valid multiprocessor computing
Thrombus did not redirect flow	Increased the viscosity of thrombus sufficiently to preclude flow	Thrombus growth into the flow volume redirects flow creating variation in shear rates and residence time
Undiagnosed errors prevented successful completion of simulation run	A tracking and error diagnostic routine was written and included in the code	Error diagnostic identified mistakes in user subroutines Tracking provides validation of successful completion of all subroutines
Code inefficiencies greatly extended runtimes	Subroutines were redesigned to avoid unnecessary loops, calculations, and storage	Reduces runtime by 50%
Values for platelet adhesion strength and thrombus coalescence contact area threshold were unknown	Necessary model parameters were estimated by statistical optimization to fit experimental results	Model predictions for embolization events match experimental results for the same conditions

4.1 Parallelize the Thrombosis and Embolization Code

The simulation of thrombosis and embolization is possible through user subroutines that handle the mechanics of platelet adhesion and embolization rates as well as criteria for thromboembolization. These subroutines were written by Goodman for the original FLUENT® (ANSYS, Canonsburg, PA) model, then later adapted to STAR-CD® and improved by Ashton, Lattin, and Frost. The STAR-CD® model took advantage of Brigham Young University's super computer in order to run on parallel processors and significantly decrease runtime. However, the authors of the model did not take into account how parallel processing might affect the subroutines.

Parallel processing in STAR-CD® uses automatic domain decomposition to divide computations evenly between available processors. Each processor is assigned an approximately equal portion of the flow volume for which it will solve the necessary transport equations for the entire simulation. Figure 4-1 shows the domain boundaries in a 64-processor simulation.

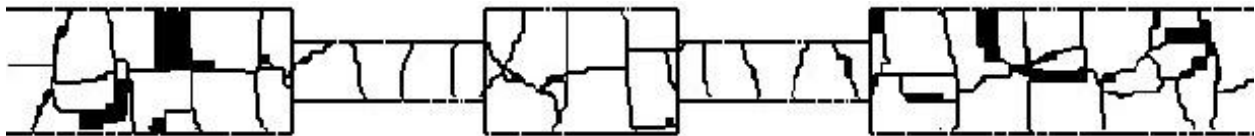


Figure 4-1 Flow cell geometry showing typical processor-domain boundaries for 64 processors

Solver algorithms ensure that transport equation solutions are consistent across all processors. This works very well for the transport equations which STAR-CD® is designed to solve. However, domain decomposition makes the prediction of thrombosis and embolization, which often requires that information local to a thrombus, more difficult. In many instances a thrombus may reside on or near a processor boundary, which will prevent the code from

gathering the necessary information about local conditions that affect thrombus growth. The thrombosis and embolization code must be written to include multi-processor communication in order to accommodate the growth, coalescence, and embolization of thrombi across processor domains.

The result of running the original code on multiple processors was that all thrombi were unintentionally restricted to the processor domain in which they were initiated, resulting in unrealistic growth patterns. To accommodate the growth, coalescence, and embolization of thrombi across processor domains, Message Passing Interface (MPI) like instructions specific to STAR-CD® were utilized. These allow for information sharing between processors at strategic points in the code, as well as calculation, storage, and communication of necessary information in a manner that ensures correct and consistent data storage across all processors.

The algorithm of the thrombosis and embolization routine both with and without the MPI implementation is outlined in the flowcharts in Figure 4-2. Sharing information across processors at the key locations indicated in the flowchart allows thrombus to grow across processor domain boundaries. An example of these multi-processor thrombi can be seen in Figure 4-3.

Besides the “share info between processors” procedures which were added to the routine, it was also necessary to store a global thrombus array on all processors that could store the sum of all necessary information gathered about a thrombus from all the processors in whose domain it exists. For example, a thrombus which bridges two processor domain boundaries will have values from each of the three processors in which it is found indicating the size of the thrombus, shear force pulling on the thrombus, adhesion strength of the thrombus to the wall, and surface areas against neighboring thrombi. To accurately discover if the thrombus meets the criteria for coalescence with a neighboring thrombi or embolization from the wall, the information from all

processors must be systematically totaled and stored at every time step so that every processor will reach the same conclusion when evaluating the thrombus against coalescence and embolization criteria.

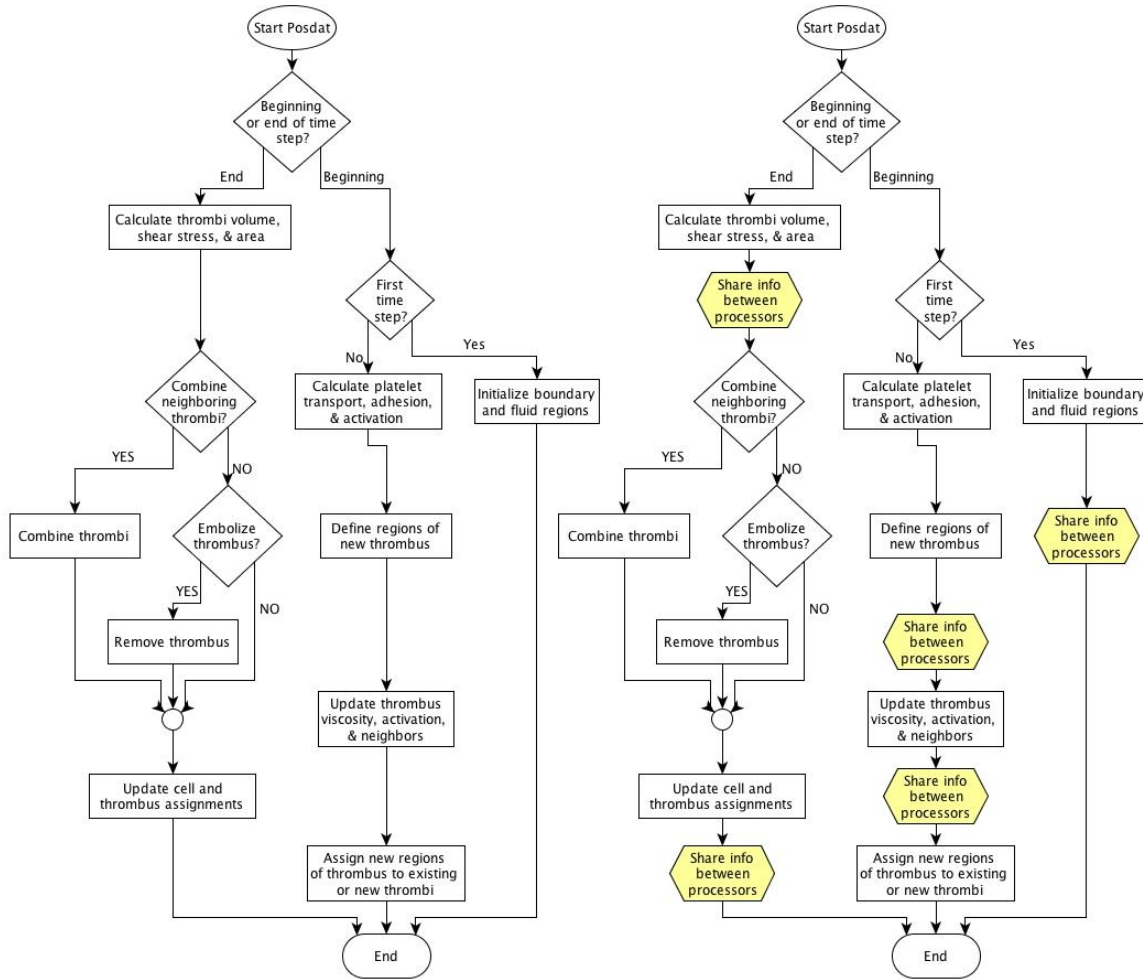


Figure 4-2 Flowchart representing thrombosis and embolization routine algorithm with (right) and without (left) MPI implementation for parallel processing

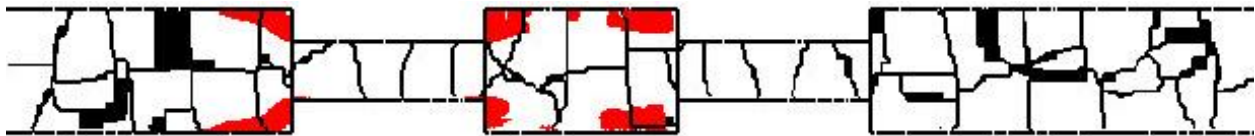


Figure 4-3 Example of successful growth of thrombi (shown in red) across processor domain boundaries

4.2 Simulate the Flow of Blood Around Thrombi

Many researchers have tried to predict thrombosis in blood-contacting devices based on the flow dynamics within the device. The problem with that method is that the event that is predicted – thrombosis – will strongly affect the conditions upon which the prediction is made – the flow field inside the device. One of the key features of this model is that as a thrombus grows into the flow field, the flow field is altered to reflect the disturbance. Flow conditions that may have originally precluded thrombus formation may now encourage it because of upstream or downstream disturbances.

The flow disturbance is created in the model by dramatically increasing the viscosity of the blood in thrombus regions. The highly viscous regions cause blood to flow around them as do physical thrombi. The flow disturbance creates areas of both stagnation and increased flow velocity and shear force. The change of a thrombus region from the constant viscosity of blood ($0.0035 \text{ Pa} \cdot \text{s}$) to a viscosity sufficiently high to stop flow ($1.0 \text{ Pa} \cdot \text{s}$ was used) is made gradually to more accurately represent the process of thrombus formation and avoid a diverging solution from the solver. (Note that it was not necessary to determine the actual viscosity of a formed thrombus, only to set the viscosity of the thrombus in the model to some value sufficiently high to divert flow.)

Lattin, Ashton, and Frost's version of the model in STAR-CD® attempted to employ the same method; however, it was not accomplished successfully due to errors in the thrombosis and embolization subroutines. The result was a slight increase in molecular viscosity, but not enough to completely stop flow in the thrombus region. This can be seen in Figure 4-4.

These figures were taken from a simulation using the Lattin-Ashton-Frost model (note that at the time their model was used the authors employed a much coarser volume mesh, which

is why the figures look very rough). Figure 4-4(a) shows thrombus growth after 102 seconds of simulation time. A thrombus, which extends into the flow field, has formed at the first sudden contraction point. If the model worked correctly, the thrombus would stop the flow and all fluid flow would be diverted around the growing thrombus. Figure 4-4(b) shows the velocity profile at the same moment as is seen in Figure 4-4(a). The velocity profile shows some disturbance of the flow in the thrombus region, but not a complete stop. The viscosity of the thrombus regions was being raised, but the jump only represented 10% of the desired viscosity change.

Correctly altering the thrombus viscosity was a key to obtaining accurate predictions from the model. Finding and correcting the error allowed us to raise the viscosity to the desired level and completely stop flow in the thrombus region. Figure 4-5 shows flow diverted around the growing thrombus at the sudden contraction point.

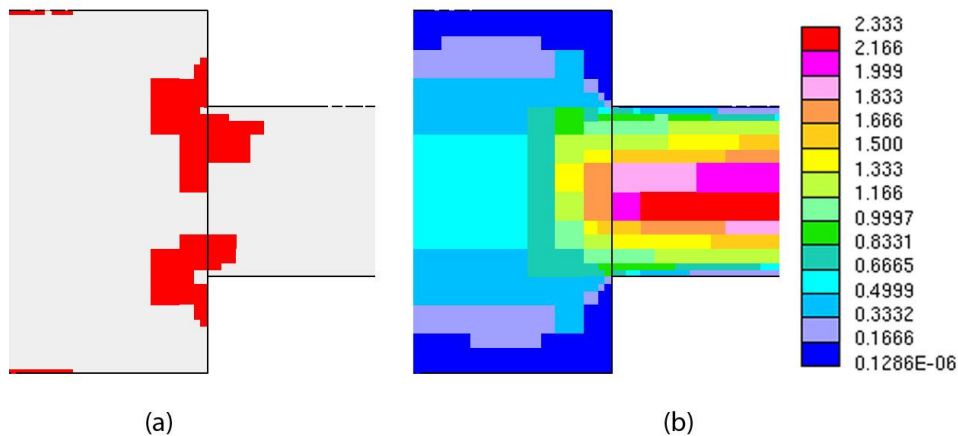


Figure 4-4 (a) Predicted thrombosis and (b) velocity profile (m/s) from the Lattin-Ashton-Frost model

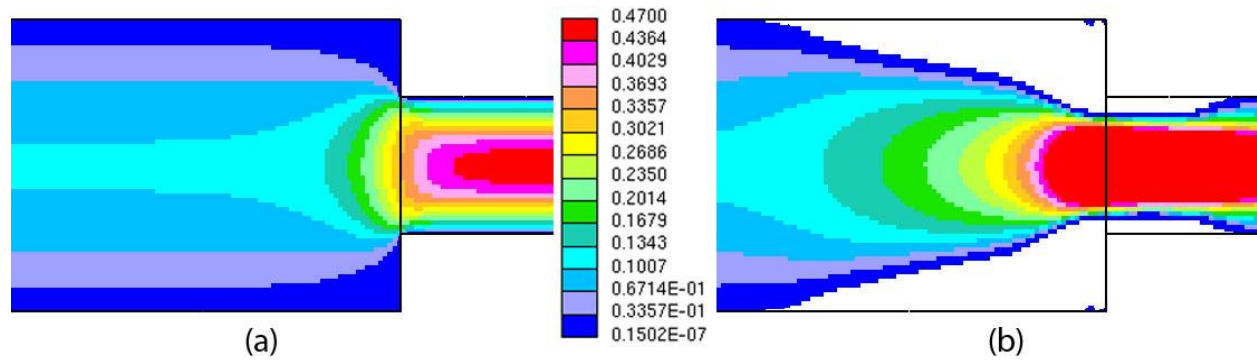


Figure 4-5 Velocity profile (m/s) (a) before and (b) after thrombus growth (thrombus indicated by white region in figure b)

4.3 Perform Detailed Tracking of Targeted Cells and Thrombi

Early in the evaluation of the Lattin-Ashton-Frost code, the simulation showed errors in thrombus growth patterns. It was easy to insert diagnostic routines to check for errors, but it was more difficult to discover the source of the error. To allow for more transparency in the execution of the model and the development of thrombi in the device, an option for executing the code with progression tracking was added. This addition performs detailed tracking of any targeted cell or thrombus and reports, at time of execution, all computational steps shown in Table 4-1 including the correct evaluation of all computational expressions, and storage and recall of values to and from the STAR-CD® solver. In the case of cells or thrombi that exist on the boundary between processors, the tracking code executes on multiple processors giving the development history as seen from all processors with information on the cell or thrombus designated. Diagnostic tools were also designed and implemented to identify and report errors periodically in the code. These improvements helped identify the problem of running the code on multiple processors before changing the subroutines for parallel processing and were especially useful in ensuring the accuracy of thrombus coalescence and embolization that occurs across multiple processors.

4.4 Optimize the Code to Minimize Runtime

Removing redundant and unnecessary calculations and data storage. Initial development of a code is often more focused on utility than efficiency. As a result, a careful review of code algorithms revealed many opportunities for runtime efficiency improvements. Figure 4-6 shows major changes made to the coalescence and embolization routine.

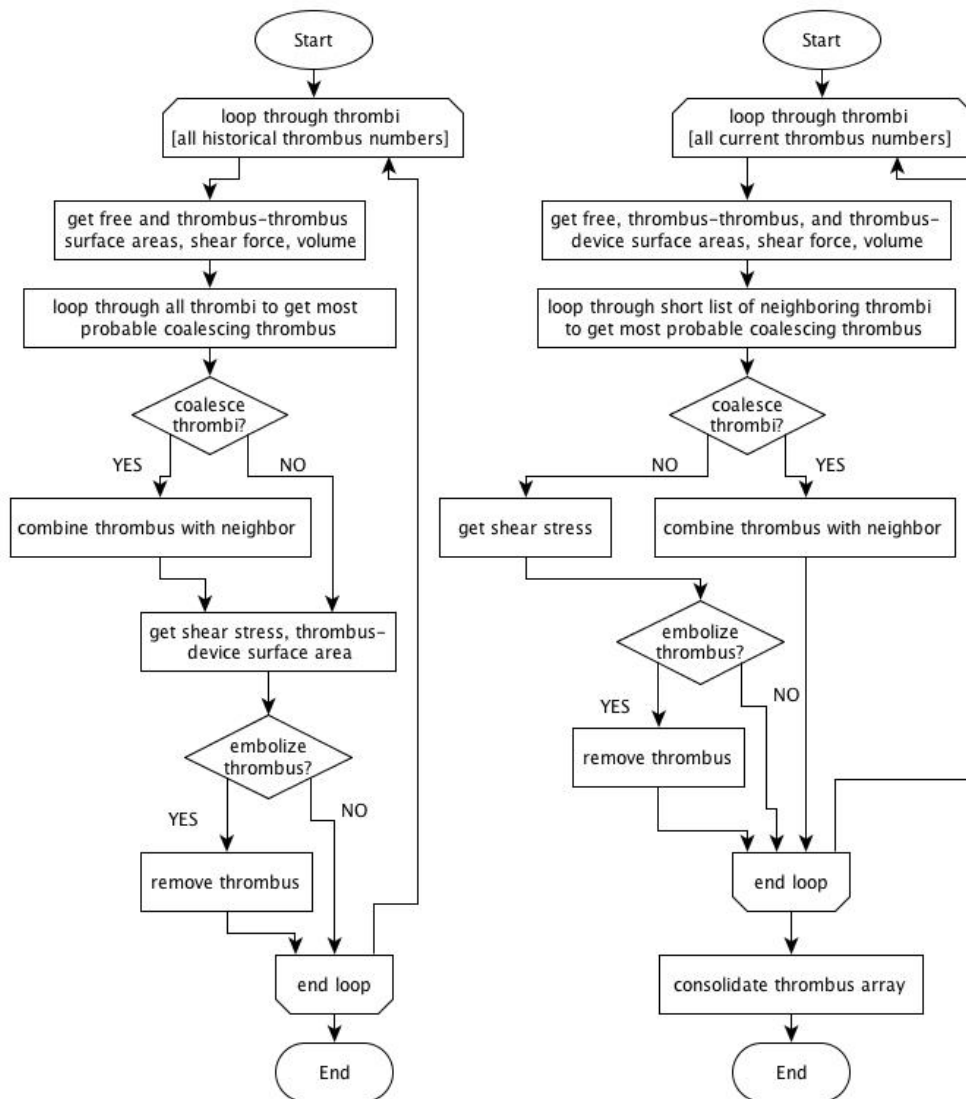


Figure 4-6 Coalescence and embolization routine (a) before and (b) after optimization

The changes include only looping through relevant thrombi for calculations (either current thrombi for outer loop, or neighboring thrombi for inner coalescence loop), and avoiding the embolization routine entirely for thrombi that have coalesced with a neighboring thrombus. These changes and others like them resulted in an approximately 50% reduction in overall runtime.

Determining an optimum mesh density. The method for solving flow fields using computational fluid dynamics involves dividing the physical domain into discrete three-dimensional cells. The properties of the fluid within each cell are assumed to be uniform, and the equation of motion is solved in agreement with the equation of continuity in each cell. Justification for the assumption of uniform conditions in a cell requires that the cell be small enough to not encompass a large change in physical properties, or velocity. The meshing guidelines established for the Lattin, Ashton, and Frost model result in cells that are too large for accurate computation (see Figure 4-7).

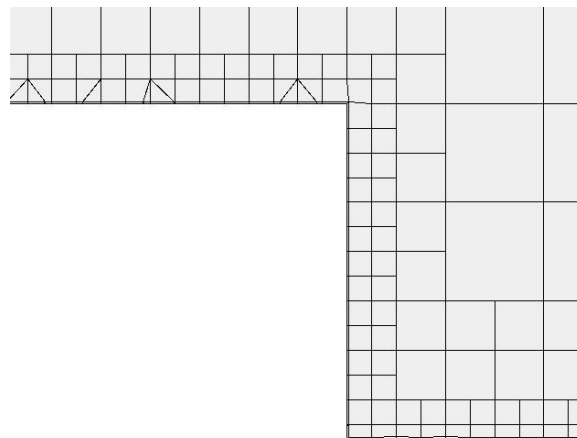


Figure 4-7 Cell stencil from Lattin-Ashton-Frost model

A series of simulations were run over a range of cell densities ($cells \times 10^{15}/m^3$) to determine the necessary grid resolution for grid-independent solutions. Figure 4-8 shows that cell densities below 2 resulted in larger average embolus volumes. To distinguish between cell densities greater than or equal to 2, thrombus locations were examined.

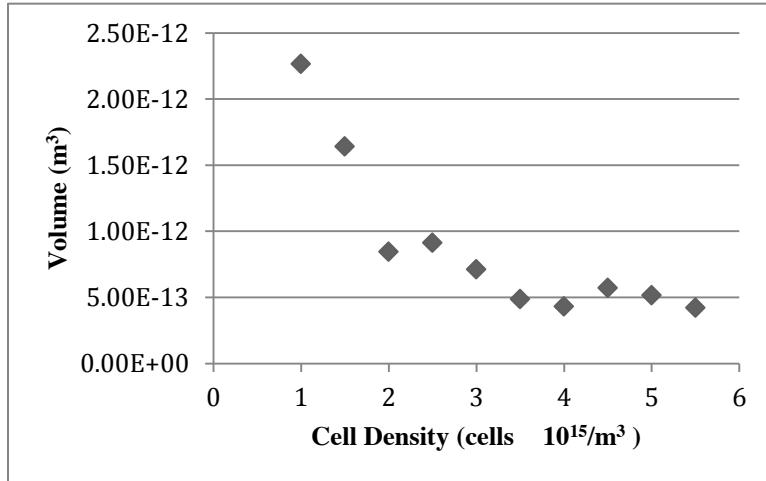


Figure 4-8 Average embolus volume vs. cell density

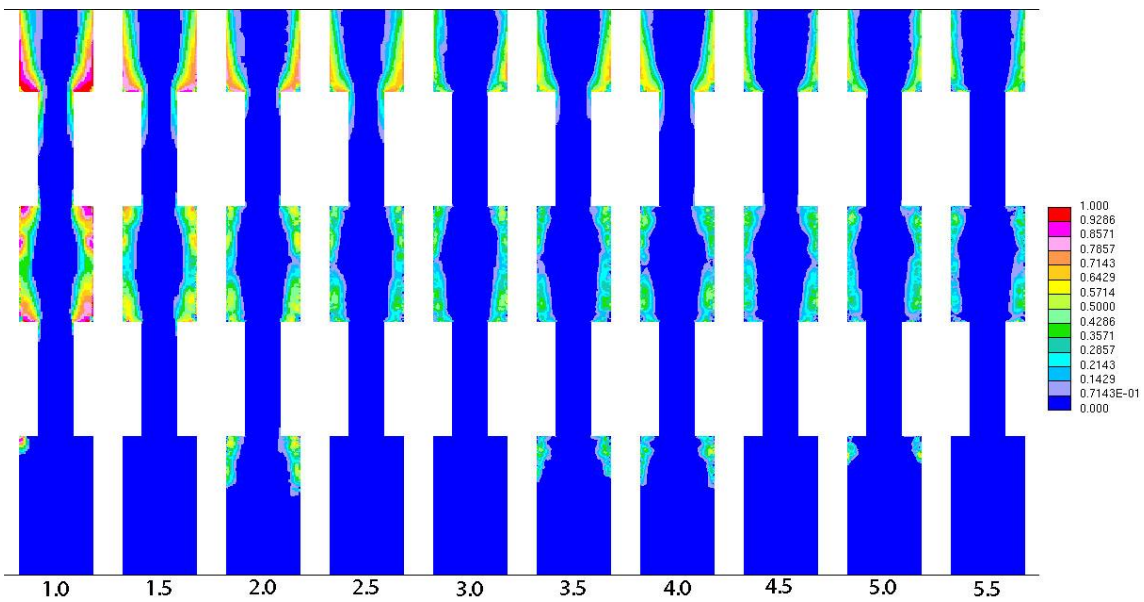


Figure 4-9 Thrombus location map

Figure 4-9 shows that the cell densities of 2 and lower have significantly more thrombus in the flow cell. Cell densities greater than 2 are all very similar in terms of the occurrence of thrombus over time. Variations exist in the appearance of thrombus in the outlet region, but the randomness of their appearance confirms that this is not the result of a specific range of cell density.

The optimal cell density will have cells small enough to justify the assumption of intra-cell uniformity but no smaller in order to optimize computation time. After examining the results of the cell density study, a cell density of 2.5 was chosen as the standard. Figure 4-10 shows that the new standard cell density is significantly finer in nature than the previously suggested mesh (Figure 4-7).

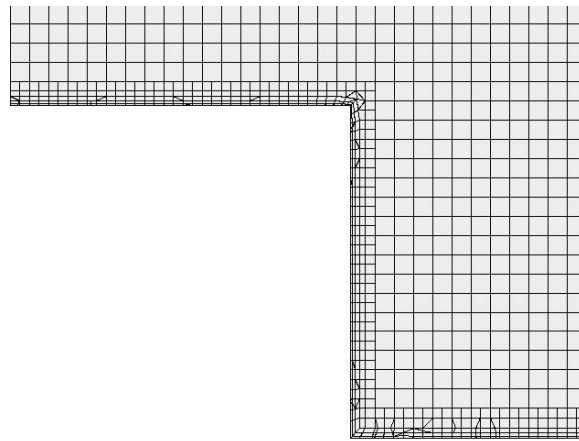


Figure 4-10 Established standard cell stencil (2.5×10^{15} cells/m³)

Determining an optimum residual tolerance for the convergence process. Optimizing the residual tolerance of a CFD model is important not only for speed of solution, but also for accuracy. The goal of optimizing the residual tolerance is to find the highest residual tolerance that achieves an accurate solution. To determine the optimum residual tolerance, tolerances of

0.008, 0.03, and 0.05 were used in otherwise duplicate simulations and the results compared (see Figure 4-11). This range was chosen based on the suggested tolerance for transient simulations using the SIMPLE solver method of 0.05 and the achievable tolerance in less than 100 iterations of approximately 0.008.

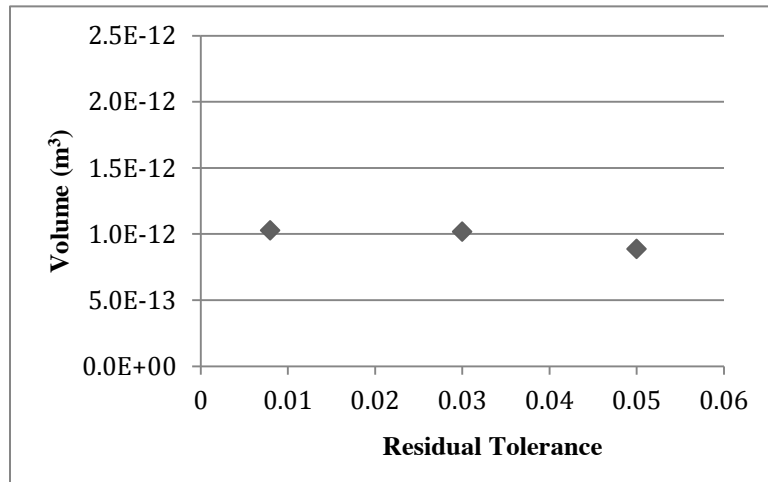


Figure 4-11 Average embolus volume vs. residual tolerance

Results show no advantage to tightening the residual tolerance to 0.008 and therefore 0.05 represents the tolerance with the highest runtime efficiency and achievable accuracy.

4.5 Estimate Necessary Model Parameters

The thrombosis and embolization routine requires the use of two unknown values: the area criterion for coalescence of neighboring thrombi and the platelet adhesion strength. Statistical methods were used to estimate these parameters by exploring the influence of variations in the coalescence criterion and adhesion strength on embolus size and number of emboli and comparing the response surface to actual experimental results. A central composite

design with additional points (at the standard distance of 1.414α from the center) to measure curvature was used to create the response surface.

The adhesion strength was estimated by Goodman[33] to be around $10 N/m^2$. Experience with the present model suggested that a slightly lower value would result in embolization patterns more consistent with experimental results. For this reason, $8 N/m^2$ was used as the center point for the adhesion strength variable. A range of $5 N/m^2$ to $11 N/m^2$ was chosen because it would include Goodman's original estimate of $10 N/m^2$.

The coalescence criterion used in the model is that a minimum threshold of percent of total surface area of a thrombus be in contact with another thrombus. If the threshold is reached or exceeded, the two thrombi coalesce. The greatest contact area percent possible with two perfect cubes is 16.7%. Because two thrombi are decidedly not perfect cubes, the critical value must be something less than 16.7%. Experience with thrombus experiments tell us that thrombi in contact with each other can be disjoined when only one thrombus is removed; therefore, we know that the critical value must be greater than zero. A center point of 8% was chosen because it is the mean of the minimum and maximum values. Values of 6% and 10% were chosen as reasonable high and low values for corner points.

The total number of emboli was chosen as the most appropriate response variable because it was assumed to be directly linked to the two independent variables in question. The computer model of thrombosis/thromboembolization was run for a 2-insert flow cell (mesh density = $2.5 \times 10^{15} \text{ cells}/m^3$) using each combination of coalescence threshold and adhesion strength as mapped in Figure 4-12 (with 5 replicate runs of the center point to test for variability).

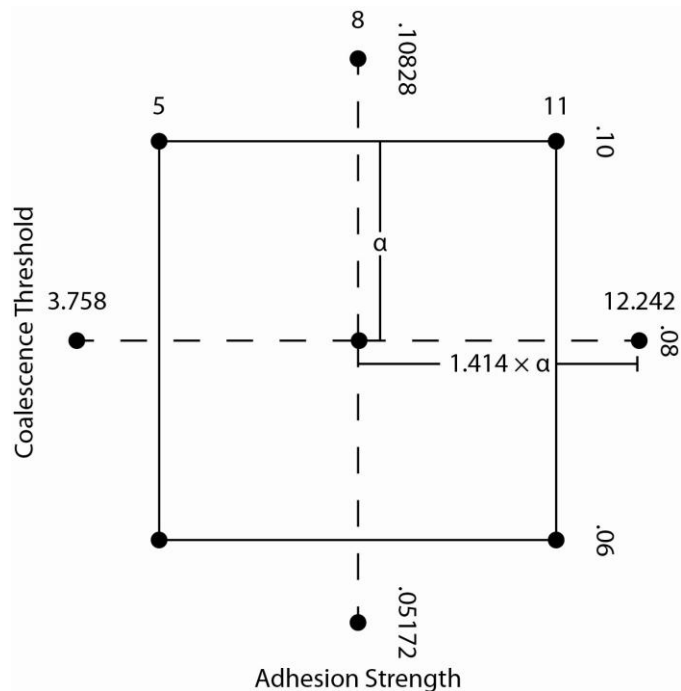


Figure 4-12 Central composite design with additional points for curvature test

From the number of emboli predicted, only those large enough to be detectable experimentally were used for comparison with experimental results. The critical size was estimated by considering experimental results from a laser light scattering system used in Dr. Solen's laboratory for microemboli detection (described below), together with a minimal detectable volume of 0.006 mm^3 taken from a calibration curve generated by Goodman[33] for that system. Results are shown in Table 4-3.

Table 4-3 Central composite design with tabulated results

Standard Order	Adhesion Strength	Coalescence % Threshold	Number of Emboli
1	5	6	122
2	11	6	145
3	5	10	102
4	11	10	115
5	3.758	8	93
6	12.242	8	132
7	8	5.172	116
8	8	10.828	87
9	8	8	118
10	8	8	114
11	8	8	118
12	8	8	118
13	8	8	114

Statistical Analysis of the results showed that the adhesion strength value, the coalescence threshold value, and the square of the coalescence threshold value were significant in predicting the total number of emboli. Fitting to a polynomial equation quantizes the effect of both adhesion strength and coalescence threshold on the number of emboli. Regression coefficients along with their standard errors are given in Table 4-4.

Table 4-4 Regression coefficients for number of emboli as a function of adhesion strength and coalescence threshold

Term	Coefficient	Standard Error of Coefficient
Constant	66.165	45.3729
Adhesion Strength	3.798	0.8862
Coalescence % Threshold	10.92	11.3842
(Coalescence % Threshold)²	-1.038	0.7066

Application of these values results in a fit equation that achieves an R² value of 81.19% and an adjusted R² value of 74.92%.

The response surface can be seen in Figure 4-13 and Figure 4-14.

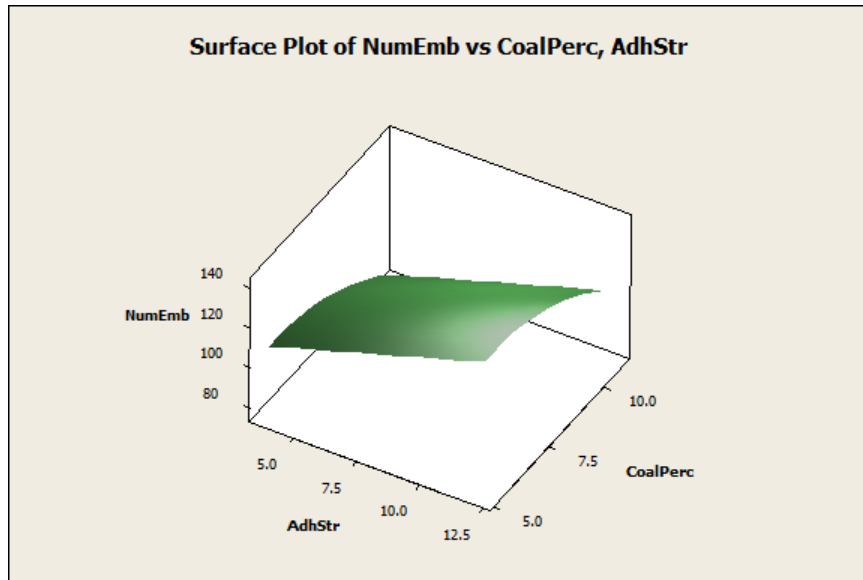


Figure 4-13 Response surface for number of emboli

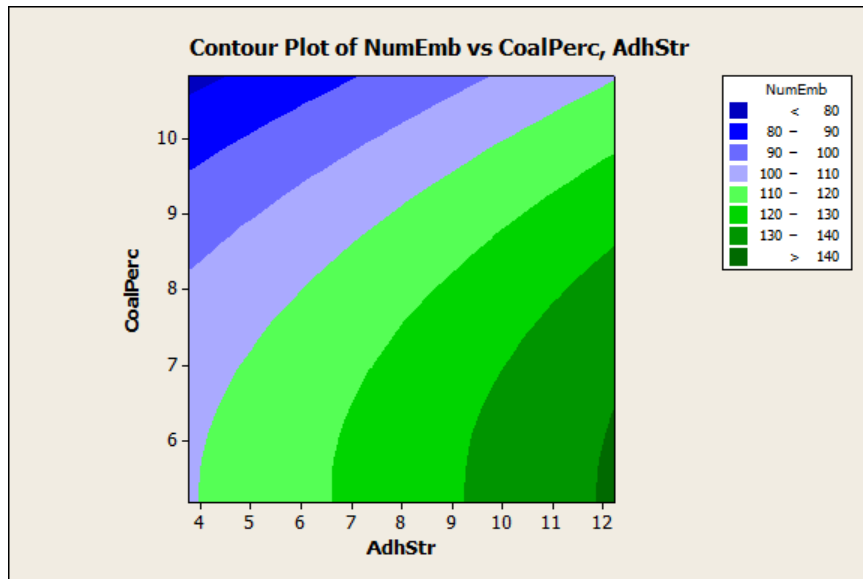


Figure 4-14 Contour plot for number of emboli

Experimental data were collected to determine the target number of emboli to estimate the adhesion strength and the threshold of contact between two thrombi to signal their coalescence for the model. Fresh human blood was passed single-pass through the flow cells. Emboli were continually monitored downstream of the flow cell using a light-scattering microemboli detector[50].

The flow cells were made from polyurethane tubing (Micro-Renathane, Braintree Scientific, Braintree, MA). Short tubing segments (1 *mm* in length, 280 μm ID, 610 μm OD) were inserted into 21-inch lengths of larger tubing (610 μm ID) spaced 1 *mm* apart. The completed flow cells, containing 2 inserts, were then rinsed by perfusing them with 5% ethanol for approximately 1 minute and then with distilled water for approximately 5 minutes.

Blood was collected by venipuncture from health human donors who had not taken medication for 2 weeks. The blood was drawn into heparin (Baxter Healthcare Corp, Deerfield, IL, 0.67 *U/mL* final concentration) in three syringes. A flow cell was then connected to each syringe via a connection in which the entrance to the flow cell tubing was inserted directly into the interior of the syringe. A multi-syringe pump was then used to generate 1.0 *mL/min* of flow from the syringes and their connected flow cells simultaneously while the entire system was maintained at 37°C.

The light scattering microemboli detector monitored the number and size of the emboli passing through the polyurethane tubing of each flow cell downstream of the inserts. Light from a laser diode (789 *nm*, 25 *mW*) was directed into the tubing, while the intensity of scattered light was continuously monitored via two detectors positioned on opposite sides of the tubing at 90° from the incident illumination. The monitored signal was fed to a computer, and the number and height of the disturbances (indicators of emboli) were recorded.

Table 4-5 Results from flow cell experiments

Experiment	Donor	Flow Cell	Laser Detector	Laser Detector	Number of Emboli
1	A	a	a	a	221
2	A	a	a	b	191
3	A	b	b	c	255
4	A	b	b	d	113
5	A	c	c	e	105
6	A	c	c	f	120
7	B	d	a	a	172
8	B	d	a	b	162
9	B	e	b	c	208
10	B	e	b	d	190
11	B	f	c	e	77
12	B	f	c	f	89
13	C	g	a	a	174
14	C	g	a	b	84
15	C	h	b	c	324
16	C	h	b	d	337
17	C	i	c	e	45
18	C	i	c	f	66

Joseph Badal, a researcher in Dr. Solen’s blood-material interactions lab at BYU performed the experiments with the help of volunteer blood donors. Nine experiments were run over three days using three blood donors; two laser channels monitored each flow cell experiment. Number of emboli data from these experiments is shown in Table 4-5.

Experiments 15 and 16 were not included in the analysis due to problems with the detection system noted by the researcher performing the experiment. Although there was a wide spread of data, the 16 data points used in the analysis were sufficient to obtain a mean number of emboli with a reasonable confidence interval. The mean number of emboli was 96.2 with a 95% confidence interval of ± 23 (see Figure 4-15). The variability in the experimental results reflects the difficulty of both classifying the behavior of blood (as it varies significantly from donor to donor) and accurately recording embolic events amid the “noise” of red blood cells and proteins that exist in the blood.

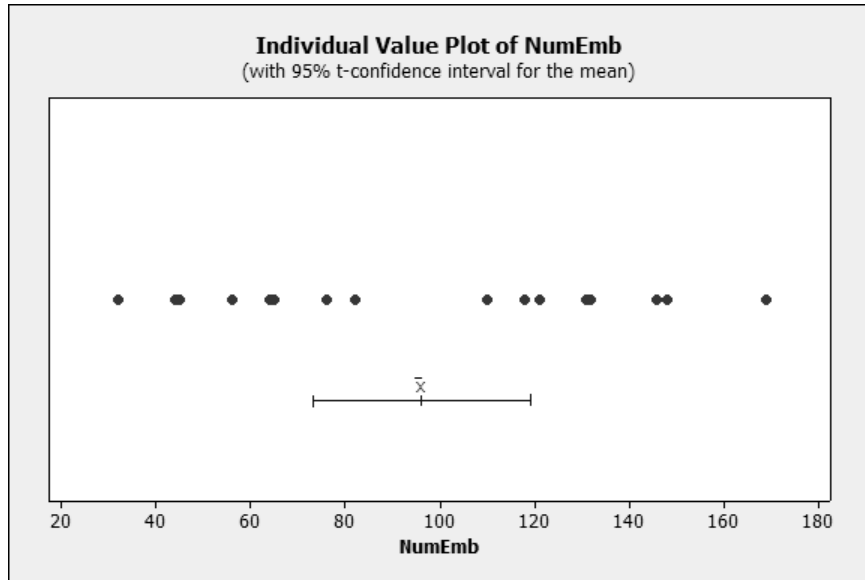


Figure 4-15 Number of emboli data with 95% confidence interval

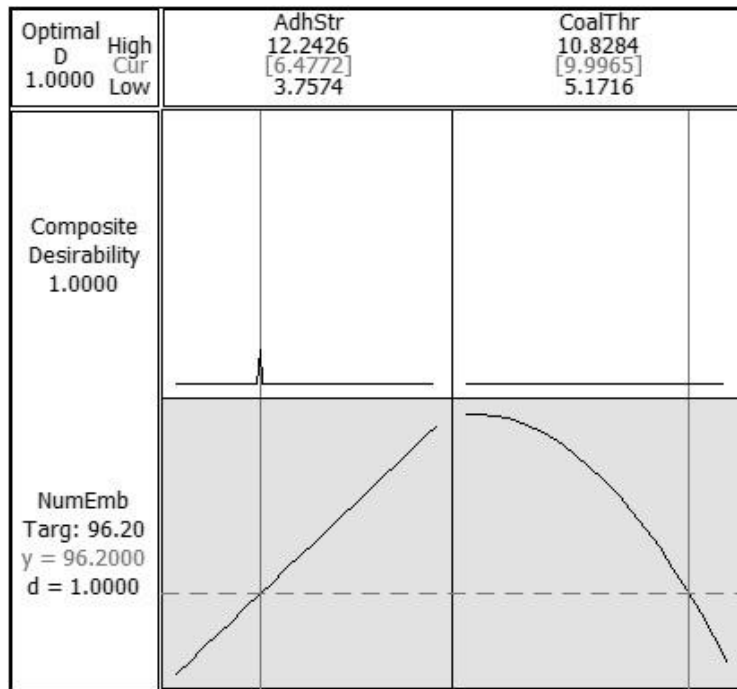


Figure 4-16 Optimization plot for adhesion strength and coalescence threshold

Fitting the mean value of number of emboli using the coefficients given in Table 4-4 provides estimates for the adhesion strength and coalescence threshold. The optimization was performed in Minitab 15® (Minitab Inc., State College, PA) and can be seen in Figure 4-16.

Table 4-6 shows the estimated values of adhesion strength and coalescence threshold to be used in the model. Using these values, the response surface predicts 96.2 emboli with a 95% confidence interval of 87.6 – 104.8.

Table 4-6 Estimated values for thrombosis model

Estimated Values	
Adhesion Strength	6.5 N/m^2
Coalescence Threshold	10.0 %

5 PREDICTING THE EFFECT OF INSERT SPACING IN THE FLOW CELL

The present model is intended to independently predict the effects of device design on thrombosis and embolization. The predictive ability of the model must be demonstrated by using it to predict the effect of variations in a device design on resulting thrombosis and thromboembolization. Since the value of those predictions can only be evaluated by comparison with experimental data, an experimental setup that allows such detailed evaluation was needed. A good candidate for such experimental work was Goodman's original flow cell, in which the design could be varied by changing the spacing between flow constriction inserts (e.g. Figure 5-1).

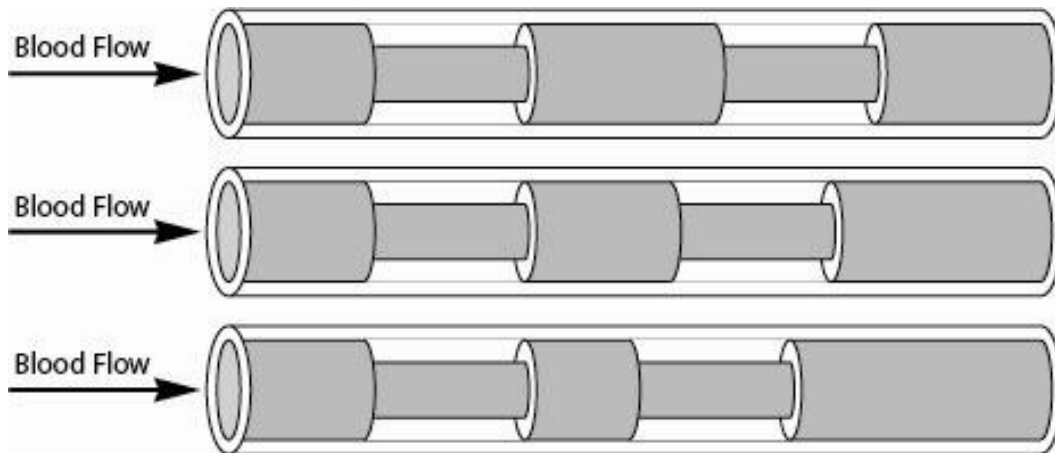


Figure 5-1 Variations on flow cell to demonstrate effect of device design

The effect of the change in the spacing between inserts must be significant enough to alter the flow of blood through the tubing. Steady-state computational fluid dynamics

simulations were performed to examine the effect of the spacing between inserts on the velocity field. Flow cell models were designed in three-dimensions with inserts spaced at 1.0 mm, 0.75 mm, and 0.5 mm. STAR-CD® was used to solve the steady-state velocity field of blood in the flow cell without the thrombosis and embolization routines using the SIMPLE solver algorithm. Comparison showed that while the inserts spaced at 1.0 mm and 0.75 mm resulted in very similar flow fields, the insert spaced at 0.5 mm gave more distinct results. In the models with inserts spaced at 1.0 mm and 0.75 mm, the flow reconnected to the wall in about the same place after exiting the first insert; however, in the model with inserts spaced at 0.5 mm, the flow exiting the first insert did not reconnect to the wall before reaching the second insert, creating a large recirculation zone (see Figure 5-2).

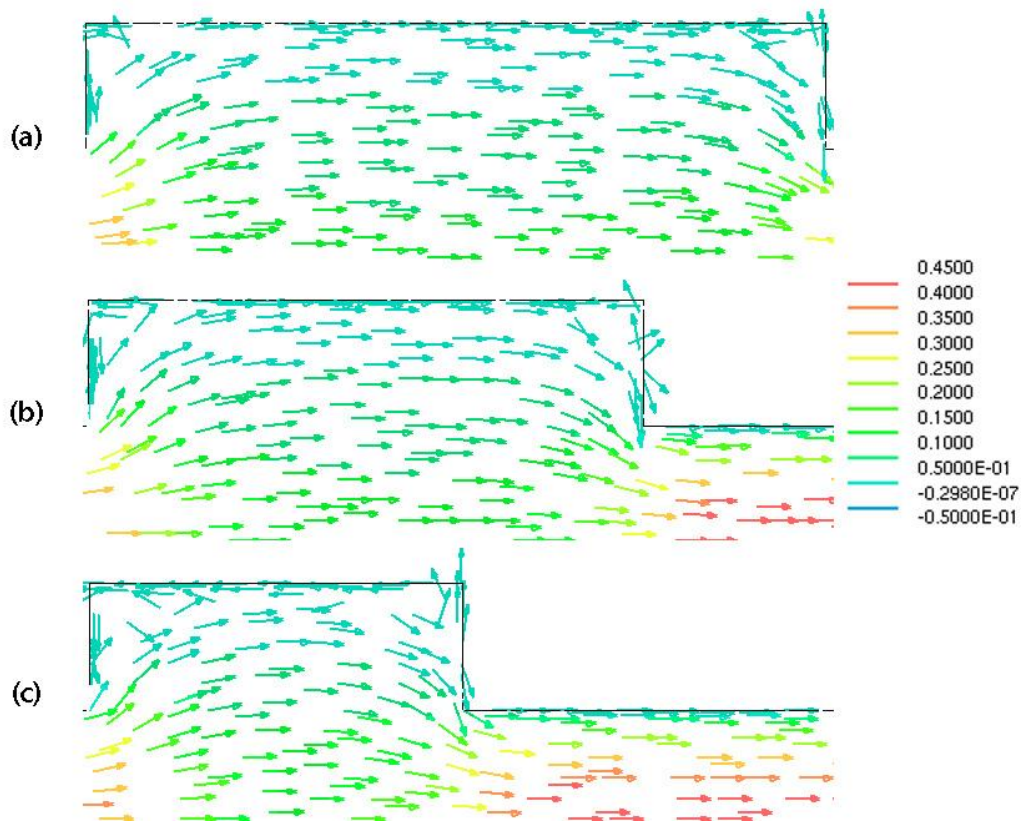


Figure 5-2 Effect of insert spacing on velocity field

Based on the results of the steady-state simulations, flow cells with inserts spaced at 1.0 mm and 0.5 mm were selected because they represent a significant change in the velocity profile in the flow cell. Results of the predictive thrombosis/thromboembolization model and the results of analogous experiments would be compared to demonstrate the model's value in the predictions of the effects of geometry variations on thrombosis and thromboembolization.

Transient simulations of thrombosis/thromboembolization were run for a 2-insert flow cell (mesh density = 2.5×10^{15} cells/m³) with inserts spaced at 1.0 mm and 0.5 mm. As in the previous analysis the total number of emboli was chosen as the primary response variable, and only the emboli large enough to be detectable experimentally (estimated to be those with volumes greater than 0.006 mm³) were used for comparison with experimental results. Other previously used response variables such as the location of thrombosis relative to the sudden expansion or contraction point[34] could have been used, but the total number of emboli was chosen to better reflect the use of transient simulation vs. the previously explored steady state CFD analysis and to take advantage of the laser detector system vs. video microscopy, which was not always available. Results are shown in Table 5-1. The adhesion strength and coalescence % threshold were set to 6.5 N/m² and 10.0% respectively. These values were determined based on the analysis described in section 4.5 of this document. Results are tabulated in Table 5-1.

Table 5-1 Summary of insert spacing computational study results

Insert Spacing (mm)	Number of Emboli	Average Embolus Volume (mm ³)	Total Embolus Volume (mm ³)
0.5	55	.076	4.17
1.0	100	.031	3.07

The results show that the flow cells with inserts spaced at 0.5 mm had fewer but larger emboli resulting in an overall increase in the total volume of emboli. It is also important to note

that at the spacing of 1.0 mm, the total number of emboli was very close to the predicted number of 96.2 and within the 95% confidence interval of 87.6 to 104.8.

A map of thrombus location shows that while the general pattern is the same, the thrombus formation in the region between inserts was observably affected by the change.

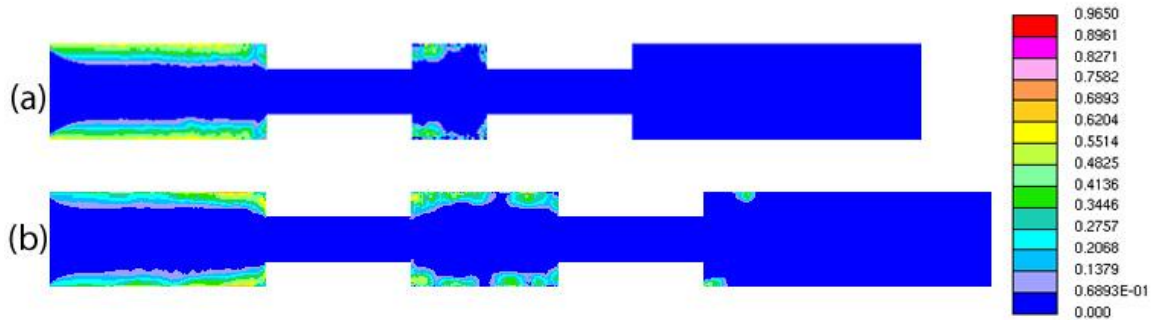


Figure 5-3 Fraction of time occupied with thrombus

Experimental data were collected to determine the effect on thrombosis and thromboembolization of altering the spacing between inserts from 1 mm as in the previous experiments to 0.5 mm. All procedures for these experiments were the same as in the previously described experiments. Joseph Badal and Kellie Bartholomew performed the experiments with the help of volunteer blood donors and provided the results shown in Table 5-2. For an accurate comparison, multiple blood samples were taken from the same donor at the same time and were run simultaneously through flow cells with inserts spaced at both 1 mm and 0.5 mm.

Analysis of the raw data shows high variability between flow cells. This is not surprising since the flow cells are made by hand. Interestingly the independent variable that appeared to have the weakest effect on the number of emboli was the insert spacing (see Figure 5-4).

Table 5-2 Experimental data from insert spacing comparison

Experiment	Experiment Set	Flow Cell	Laser Detector	Spacing	Number of Emboli
1	A	A	ch0	1.0	37
2	A	A	ch1	1.0	39
3	A	B	ch3	0.5	76
4	A	B	ch4	0.5	45
5	A	C	ch6	1.0	105
6	A	C	ch7	1.0	152
7	B	D	ch0	0.5	29
8	B	D	ch1	0.5	32
9	B	E	ch3	1.0	130
10	B	E	ch4	1.0	91
11	C	F	ch3	0.5	114
12	C	F	ch4	0.5	132
13	C	G	ch6	1.0	24
14	C	G	ch7	1.0	28
15	D	H	ch0	1.0	205
16	D	H	ch1	1.0	141
17	D	I	ch6	0.5	141
18	D	I	ch7	0.5	136

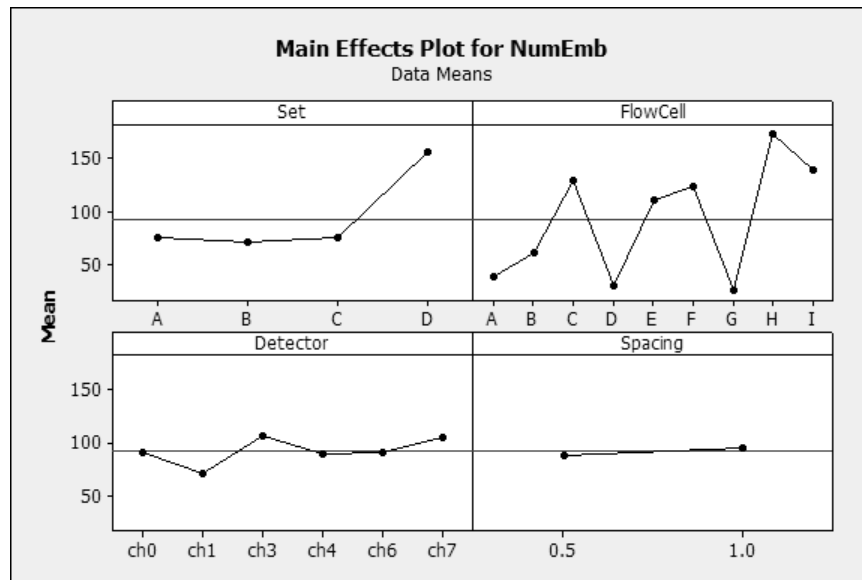


Figure 5-4 Main effect of all independent variables in spacing experiments

To isolate the effect of spacing from donor-to-donor variability, each experimental set (involving a single donor) was analyzed separately. This is accomplished by comparing the average effect of insert spacing for a single donor. Results are tabulated in Table 5-3.

Table 5-3 Effect of insert spacing on average number of emboli (averaged from the data in Table 5-2)

Experiment Set	0.5 mm	1.0 mm
A	60.5	83.3
B	30.5	110.5
C	123.0	26.0
D	138.5	173.0

The data shows a wide spread in the number of emboli for both the 1 mm and 0.5 mm flow cells and that the effect of the insert spacing was not consistent among experiments. These results reflect the difficulty of categorizing the behavior of blood as it varies widely between donors and for a single donor depending on diet, temperament, and health, and also the difficulty in manufacturing the flow cells consistently and accurately detecting the emboli among the many biological species that exist in blood. However, because the current model is only intended to predict the average effect, the analysis may still be useful. A paired t-test was performed to examine the effect of the insert spacing (see Figure 5-5). Results show an estimated population mean difference of -10.1 with a 95% confidence interval of -130.2 to 110.1. Clearly, the data do not offer any statistically significant effect.

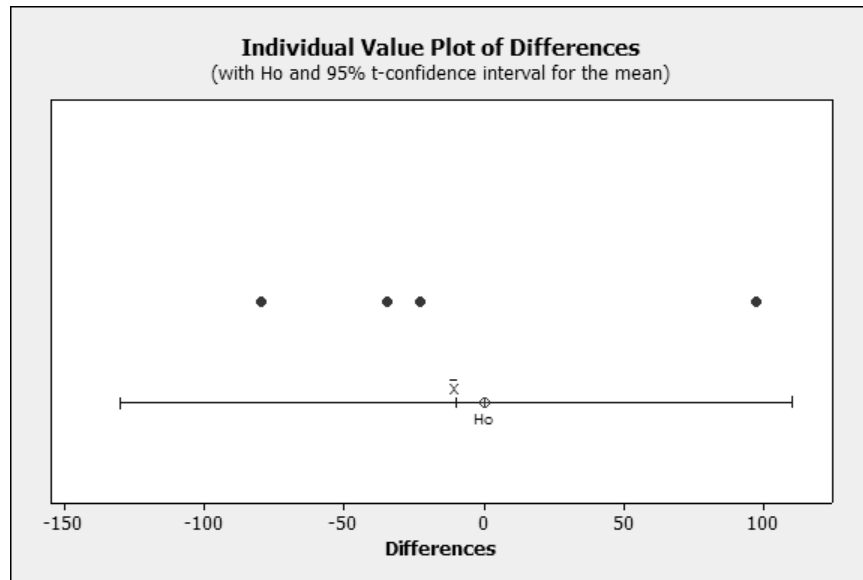


Figure 5-5 Paired t-test of average number of emboli as effected by insert spacing

Further experiments may provide a better description of the actual effect of varying the spacing between inserts; however, it is more likely that the change in insert spacing simply did not exert a significant effect on the thromboembolization potential. A more conclusive demonstration of the model would require a more significant difference between the geometries to be compared.

The comparison did show general agreement on the location of emboli initiation. However, contrary to experimental results, the model did not predict thrombosis in the constricted sections of the flow cell. The reader is referred to Chapter 7 of this document for further analysis of this observation.

6 DEMONSTRATE THE MODEL'S USE FOR DESIGNING A HEMODIALYSIS CATHETER

As a demonstration of the model's use in the design of blood-contacting devices, the simulation was applied to a Mahurkar blood dialysis catheter. A section of the catheter was measured and model in SolidWorks®. Figure 6-1 shows the catheter model and it's subsequent insertion into a flow volume representative of the superior vena cava.

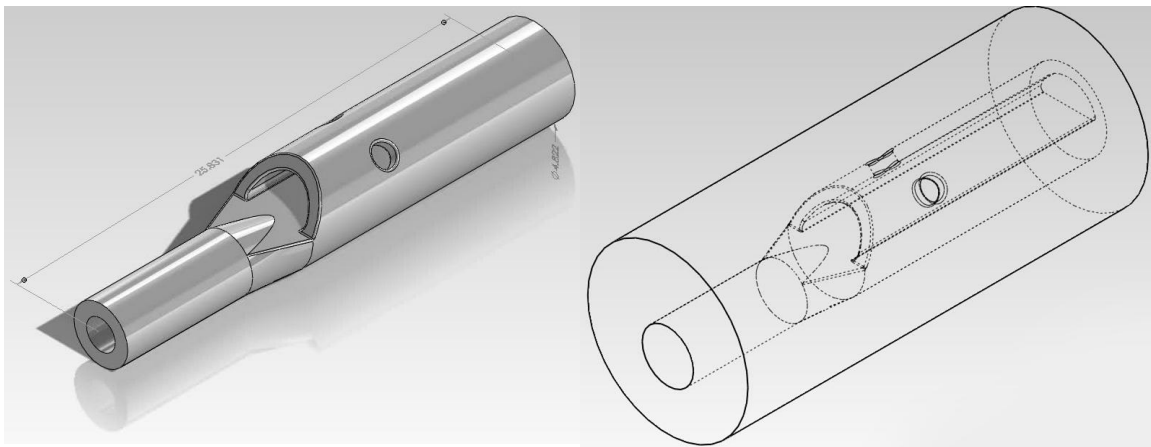


Figure 6-1 Standard Mahurkar blood dialysis catheter design

The catheter is designed to take blood from the patient through the semicircular inlet and two circular perforations. The blood then goes out to the dialysis machine and after being cleaned is returned through the catheter outlet located beneath and downstream of the inlet section. The present analysis deals only with the thrombosis and embolization at the catheter inlet section, so the outlet will not be modeled here.

The blood catheter has an outside diameter at the largest point of approximately 4.8 mm, and the superior vena cava (the common insertion point of the catheter) has an average inside diameter of approximately 1.5 to 2 cm. This resulted in a large percentage of the computational volume being empty space. Reducing the diameter of the blood vessel allowed us to drastically reduce the total computational volume without affecting the simulations results. Though care was taken to appropriately scale the fluid velocities to maintain consistency with physiological conditions and the no-slip condition was relaxed for the constricted blood vessel wall to avoid its interference with the flow profile in the vicinity of the blood catheter.

A summary of thrombus history in the device after 330 seconds of simulation time is shown in Figure 6-2.

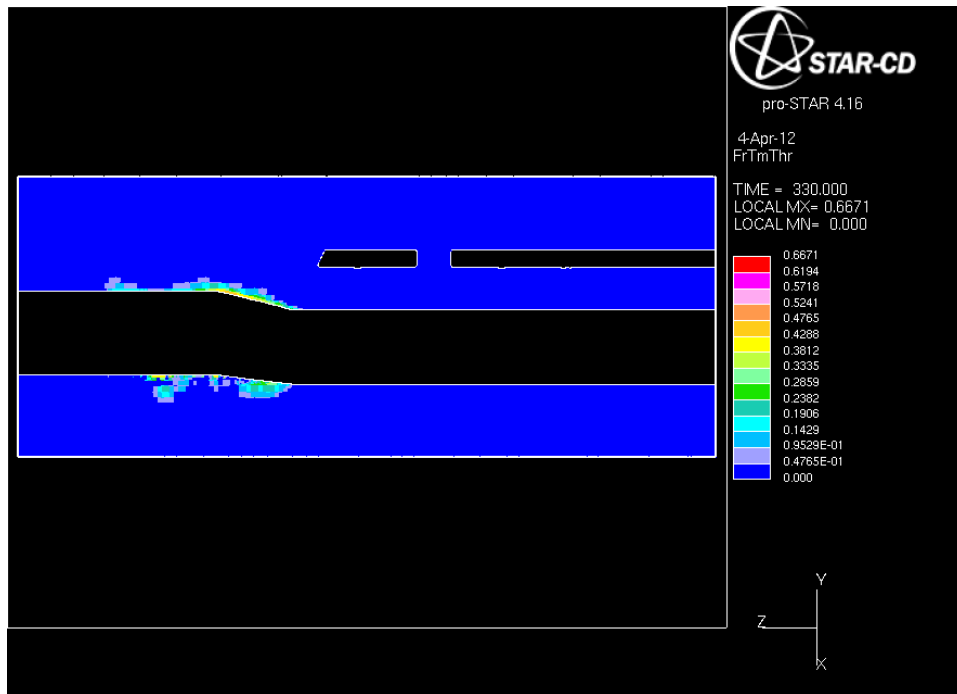


Figure 6-2 Thrombus history in standard Mahurkar catheter after 330 seconds

Originally, it was believed that the inlet holes, which force flow perpendicular to the bulk flow path, would cause activation and deposition of platelets on the inner catheter wall. Results show however, that at least while the device is operating, the shear rates in the inlet portion of the catheter are too high to allow thrombus to form. A discussion of the critical shear rate used in the program can be found in Chapter 7. Instead, we see thrombus forming in a stagnation zone where the bulk flow splits between continuing downstream and proceeding up the catheter inlet (see Figure 6-3).

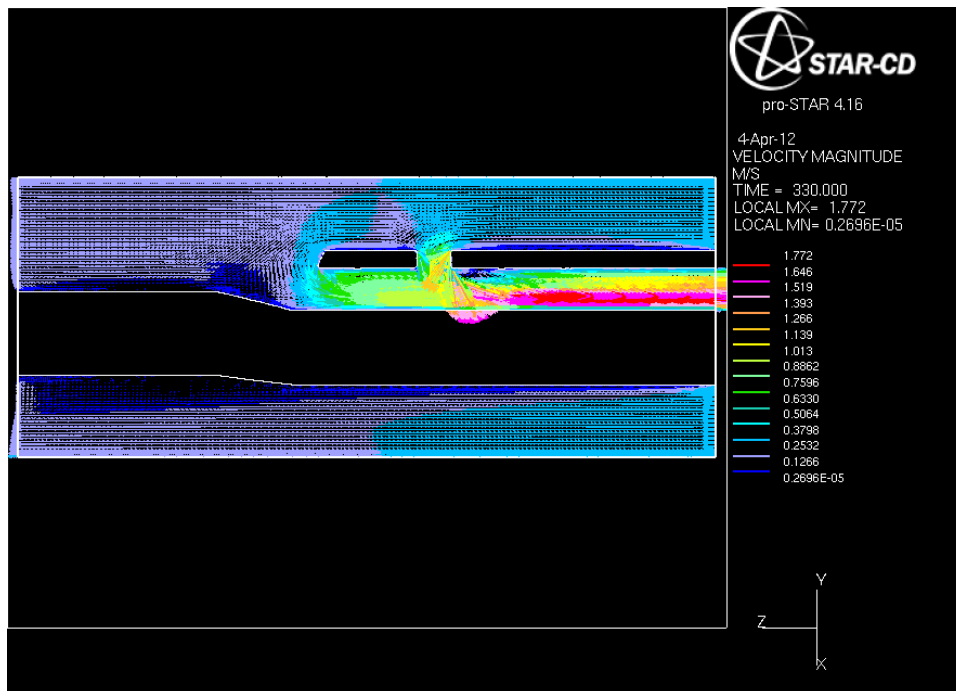


Figure 6-3 Velocity vector field in standard Mahurkar after 330 seconds

A variation on the standard Mahurkar design was created which, it was believed, would decrease thrombosis and embolization by redirecting the flow from the circular inlet holes to an angle significantly less than 90°. The design can be see in Figure 6-4.

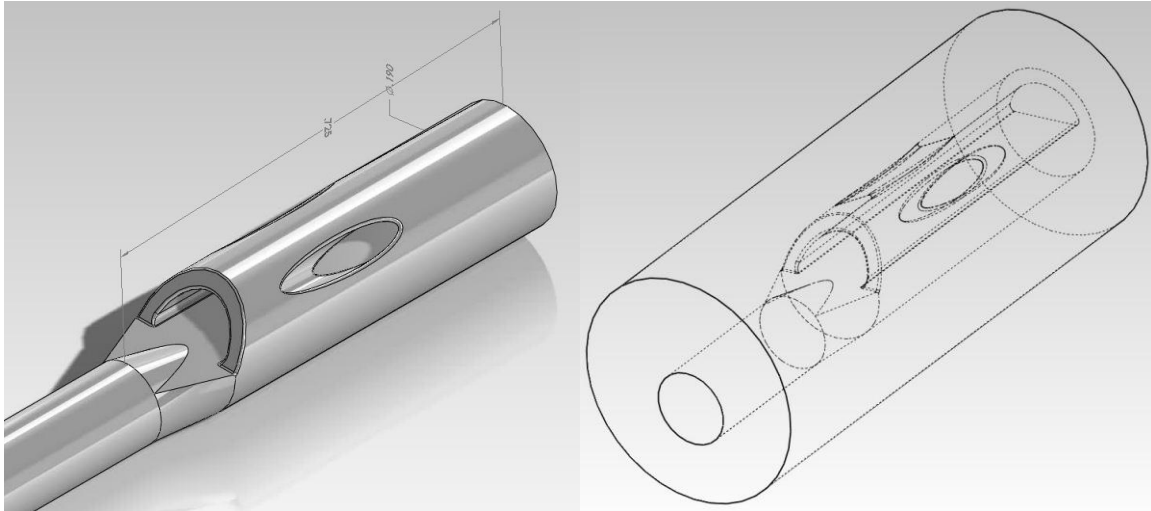


Figure 6-4 Augmented Mahurkar blood dialysis catheter design

The results of applying the simulation to this new geometry can be seen in Figure 6-5.

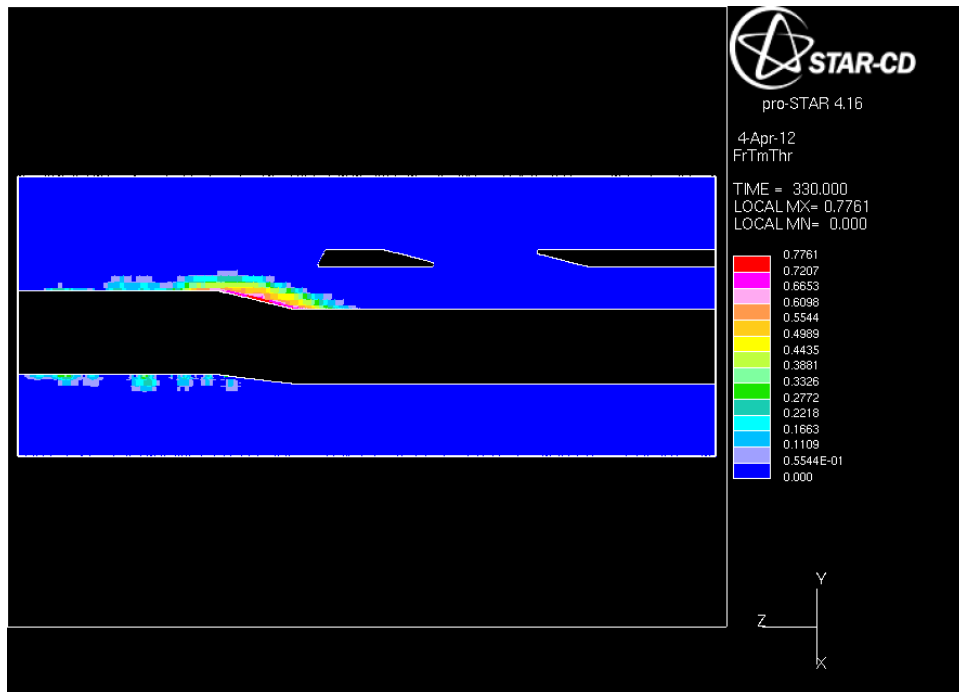


Figure 6-5 Thrombus history in augmented Mahurkar catheter after 330 seconds

The augmented design actually predicted more thrombus formation. It is believed that this is due to more of the total dialysis inlet flow coming through the circular perforations versus the semicircular inlet resulting in lower flow rates in that region and longer residence times. The velocity vector field shown in Figure 6-6 confirms this.

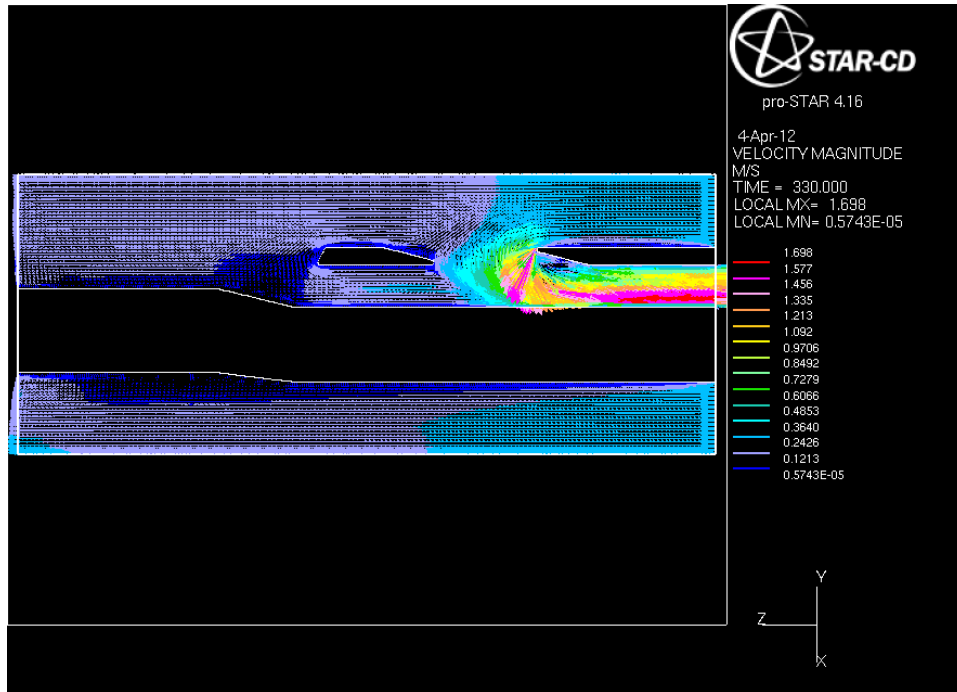


Figure 6-6 Velocity vector field in augmented Mahurkar after 330 seconds

While the augmented catheter design did not decrease the risk of thrombosis and thromboembolization as was believed, the value of using a simulation for testing variations on a device for their thrombogenic potential is apparent through this exercise. A similar study involving in vitro or in vivo experiments would have required significantly more time and money for presumably the same or similar results.

7 CONCLUSION AND RECOMMENDATIONS

A multi-processor model of thrombosis and embolization was developed to aid in the development of blood-contacting devices that minimize the risk of thrombosis and thromboemboli. Previous work by Daniel Lattin, Jeff Ashton, and Denzel Frost formed the basis for the model. The model was successfully written to take advantage of parallel processing. Successful simulation of the flow of blood around developing thrombi resulted in stagnation and high shear regions in the flow field, which greatly affect the rates of thrombosis and embolization. A system of simulation tracking was developed to allow for detailed reporting of real-time calculations and thrombus development. Removing unnecessary loops and calculations further reduced runtime (approximately 50%). The model required the estimation of two key parameters: the adhesion strength of platelets to the device wall and the threshold criterion for coalescence of two neighboring thrombi. These parameters were estimated using statistical analysis of a central composite design and fitting to experimental data. A preliminary validation of the model suggests that the computational model could predict the effect of design variations on the risk of thromboembolic events.

Further experiments are needed to adequately compare experimental observation with the computational results. It will then be possible to make adjustments to an actual blood-contacting device and examine the predicted effects on thrombosis and embolization. Testing a design in this way will reduce the time and cost of designing such devices and allow for better optimization of the geometry to reduce the risk of thrombosis and embolization.

There are opportunities for continuing work on the design of the model algorithms. Improvement could be made to the treatment of the effect of shear stress on platelets. The model presented here does not take into account a platelet's history. The additive effect of shear stress as a platelet moves along a flow path could be incorporated to better represent the possibility for shear-mediated activation of platelets. However, some initial analysis suggests that at least in the flow cell, shear activation is unlikely due to short residence times.

One present concern is that in the flow cell experiments it is not unusual to see thrombus growing in the constricted section of the flow volume, while this is not seen in the computational simulation. This is due to a shear rate threshold in the computational simulation above which wall thrombi are not allowed to initiate. Corbett et al. [34] reported this value to be 54 s^{-1} , but also reported a previously established value of $10 - 20 \text{ s}^{-1}$ (the present model uses 20 s^{-1}) pointing to several published works [46-49]. The simulation predicts shear rates in that section above 5500 s^{-1} . Given the extremely high shear rates in the constricted section, which would make it difficult for platelets to adhere to the wall, it is possible that the thrombi seen there started in a lower shear region and subsequently either grew or were pushed by flow forces into the constricted section. However it is also possible that the application of a shear rate threshold for thrombosis leads to the inconsistency. Some recent investigation into the shear rate threshold for thrombosis suggests that: 1) reported values may not be applicable to our transient simulation (Corbett et al.'s experiments differ from hours in that they incorporated a much higher heparin concentration, bovine blood, and a blood loop, and were evaluated at the end of a 2 hour experiment), and 2) such a threshold may not be needed in a computational model because there is evidence to support that such a threshold may not exist in physiological systems (the works

cited by Corbett et al. are inconsistent in their methods and unclear in their reporting of a threshold for thrombosis). Further investigation into this threshold parameter is needed.

Enhancements to the model could include allowing the model to predict the deformation or rolling of thrombi as directed by flow forces; the present model does not allow for any movement of thrombus after formation except in the case of removal due to shear forces. This movement is often seen in experiments as thrombi deform, creep, or roll along the wall in the direction of flow. Computationally this could be accomplished by evaluating not just the total force on a thrombus with each time step but the vector forces and their potential for pushing, pulling, or detaching the thrombus or a part of the thrombus. This would introduce many complications in the definition of cells as either fluid or thrombus. As the simulated thrombus moved in the flow field, cells would constantly need to be redefined at the thrombus-fluid boundary. However, there is little to be gained by this possible extension of the current model as it is meant to predict the risk of thrombosis and embolization as predicted by the flow geometry. The transient movement of a formed thrombus is believed to contribute little to the overall thrombogenic activity of a device.

Perhaps a better candidate for improvement would be altering the inlet velocity to more closely reflect in vivo conditions for devices on the arterial side of a patient's circulatory system (e.g. coronary stents, arterial grafts, etc.). The pulsatile nature of blood flow, for example, could be included and may prove to have a significant effect on the processes of thrombosis and embolization due to longer residence times and circular stress patterns.

Another possible improvement to the model would be to continue to track the movement of thrombi after embolization until they exit the flow volume. In the present model thrombi that meet the criteria for embolization are removed instantly from the flow volume, with no effect on

the downstream characteristics. However, it is possible that as the thrombi flow downstream they may have a continued effect on thrombosis or embolization by affecting shear rates around or actually coming into contact with neighboring thrombi.

A final recommendation for the further improvement of the model is to apply a temporally dynamic cell stencil. Such a system would allow the division of cells to change with each time step, as governed by the size of cells locally necessary to justify the assumption of uniformity of all physical properties and accuracy of derivative evaluation within a cell. In bulk flow areas, larger cells would be used to speed computation time while locations near a boundary or recirculation zone would be populated with the very small cells necessary for accurate computation. While this would necessitate the significant work of freeing the simulation from the STAR-CD® framework, the benefits in computation time and accuracy may outweigh the costs.

While this work represents a substantial contribution to the prediction of thrombosis and thromboembolization in blood-contacting devices, it is clear that there is still much work to be done. This work provides insight into the factors affecting thrombosis and embolization, especially the effect of fluid dynamics, and shows promise for effective use in the design of blood-contacting devices that will more successfully limit the risk of thrombosis and thromboembolization.

REFERENCES

1. Walterskirchen Mv: **The U.S. Market for Medical Devices - Opportunities and Challenges for Swiss Companies.** *Swiss Business Hub USA* 2004.
2. Panescu D: **Medical device development.** *Conference Proceedings: Annual International Conference Of The IEEE Engineering In Medicine And Biology Society IEEE Engineering In Medicine And Biology Society Conference 2009*, **2009**:5591-5594.
3. Akutsu T, Takagi H: *Trans Am Soc Art Intern Organs* 1969, **15**:55.
4. Hugues J: *Thrombos Diasthes Haemorrh* 1959, **3**:177.
5. Porter KA: *Journal of clinical Path Suupl* 1967, **20**:518.
6. Schoen FJ: **Host Reactions to Biomaterials and Their Evaluation.** In: *Biomaterials Science: An introduction to Materials in Medicine.* Edited by Ratner BD, Hoffman AS, Schoen FJ, Lemons JE, 2 edn. San Diego: Elsevier Academic Press; 2004: 293-354.
7. Sorensen EN, Burgreen GW, Wagner WR, Antaki JF: **Computational simulation of platelet deposition and activation: I. Model development and properties.** *Annals Of Biomedical Engineering* 1999, **27**(4):436-448.
8. Meksyn D: **New methods in laminar boundary- layer theory:** New York: Pergamon Press; 1961.
9. Lighthill MJ: *Proc R Soc, A* 1950, **202**:359.
10. Reynolds LO, Newren WH, Jr., Scolio JF, Miller IF: **A model for thromboembolization on biomaterials.** *Journal Of Biomaterials Science Polymer Edition* 1993, **4**(5):451-465.
11. Gorbet MB, Sefton MV: **Biomaterial-associated thrombosis: roles of coagulation factors, complement, platelets and leukocytes.** *Biomaterials* 2004, **25**(26):5681-5703.
12. Dutton RC, Webber AJ, Johnson SA, Baier RE: *Journal of biomedical materials research* 1969, **3**:13.
13. Macmillan DC: *Nature* 1966, **211**:140.
14. Richardson PD: *Bulletin of the New York Academy of Medicine* 1972, **48**:379.

15. Hampton JR, Mitchell JR: **Abnormalities in platelet behaviour in acute illnesses.** *British Medical Journal* 1966, **1**(5495):1078-1080.
16. Born GV: **Observations on the change in shape of blood platelets brought about by adenosine diphosphate.** *The Journal Of Physiology* 1970, **209**(2):487-511.
17. Friedman LI, al. e: *Transactions - American Society for Artificial Internal Organs* 1970, **16**:63.
18. Hanson SR, Harker LA: **Blood Coagulation and Blood-Materials Interactions.** In: *Biomaterials Science, An introduction to Materials in Medicine.* Edited by Ratner BD, Hoffman AS, Schoen FJ, Lemons JE. San Diego: Academic Press; 1996: 193-199.
19. Hanson SR, Harker LA, Ratner BD, Hoffman AS: **In vivo evaluation of artificial surfaces with a nonhuman primate model of arterial thrombosis.** *The Journal Of Laboratory And Clinical Medicine* 1980, **95**(2):289-304.
20. Mustard JF: **The Fate of Thrombi.** In: *Thrombosis.* Edited by Sherry S, Brinkhous KM, Genton E, Stengle JM. Washington, D.C.: National Academy of Sciences; 1969: 496-505.
21. Richardson PD: **Effect of Blood Flow Velocity on Growth Rate of Platelet Thrombi.** *Nature* 1973, **245**:103-104.
22. Begent N, Born GVR: *Nature* 1970, **227**:926.
23. Ruckensein E, Marmur A, Gill WN: **Growth Kinetics of Platelet Thrombi.** *Journal of Theoretical Biology* 1977, **66**:147-168.
24. Benis AM, Nossel HL, Aledort LM, Koffsky RM, Stevenson JF, Leonard EF, Shiang H, Litwak RS: **Extracorporeal model for study of factors affecting thrombus formation.** *Thromb Diath Haemorrh* 1975, **34**(1):127-144.
25. Dejardin P, Tenhove P, Yu XJ, Brash JL: **COMPETITIVE ADSORPTION OF HIGH-MOLECULAR-WEIGHT KININOGEN AND FIBRINOGEN FROM BINARY-MIXTURES TO GLASS-SURFACE.** *Langmuir* 1995, **11**(10):4001-4007.
26. Leduc C, Depaola N, Konath S, Vroman L, Leonard EF: **ADSORPTION OF PROTEINS OUT OF PLASMA ONTO GLASS FROM A SEPARATED FLOW.** *J Biomater Sci-Polym Ed* 1994, **6**(7):599-608.

27. Lu CF, Nadarajah A, Chittur KK: **A COMPREHENSIVE MODEL OF MULTIPROTEIN ADSORPTION ON SURFACES**. *Journal of Colloid and Interface Science* 1994, **168**(1):152-161.
28. Mandrusov E, Yang JD, Pfeiffer N, Vroman L, Puszkin E, Leonard EF: **Kinetics of protein deposition and replacement from a shear flow**. *Aiche J* 1998, **44**(2):233-244.
29. Slack SM, Horbett TA: **CHANGES IN THE STRENGTH OF FIBRINOGEN ATTACHMENT TO SOLID-SURFACES - AN EXPLANATION OF THE INFLUENCE OF SURFACE-CHEMISTRY ON THE VROMAN EFFECT**. *Journal of Colloid and Interface Science* 1989, **133**(1):148-165.
30. Wojciechowski P, Brash JL: **The Vroman effect in tube geometry: the influence of flow on protein adsorption measurements**. *Journal Of Biomaterials Science Polymer Edition* 1991, **2**(3):203-216.
31. Marmur A, Cooper SL: **A model for the deposition and detachment of proteins and platelets on biomaterials**. *Journal of Colloid and Interface Science* 1982, **89**(2):458-465.
32. Wilson RS, Marmur A, Cooper SL: **A model of deposition and embolization of proteins and platelets on biomaterial surfaces**. *Annals Of Biomedical Engineering* 1986, **14**(4):383-400.
33. Goodman PD, Barlow ET, Crapo PM, Mohammad SF, Solen KA: **Computational model of device-induced thrombosis and thromboembolism**. *Annals Of Biomedical Engineering* 2005, **33**(6):780-797.
34. Corbett SC, Ajdari A, Coskun AU, N-Hashemi H: **In vitro and computational thrombosis on artificial surfaces with shear stress**. *Artificial Organs* 2010, **34**(7):561-569.
35. Fraser KH, Zhang T, Taskin ME, Griffith BP, Wu ZJ: **Computational fluid dynamics analysis of thrombosis potential in left ventricular assist device drainage cannulae**. *ASAIO Journal (American Society For Artificial Internal Organs: 1992)* 2010, **56**(3):157-163.
36. Friedrich P, Reininger AJ: **Occlusive thrombus formation on indwelling catheters: in vitro investigation and computational analysis**. *Thrombosis And Haemostasis* 1995, **73**(1):66-72.

37. Okamoto E, Hashimoto T, Inoue T, Mitamura Y: **Blood compatible design of a pulsatile blood pump using computational fluid dynamics and computer-aided design and manufacturing technology.** *Artificial Organs* 2003, **27**(1):61-67.
38. Sotiropoulos F, Borazjani I: **A review of state-of-the-art numerical methods for simulating flow through mechanical heart valves.** *Medical & Biological Engineering & Computing* 2009, **47**(3):245-256.
39. Wiwanitkit V: **In vitro, computational, and in vivo thrombosis.** *Artificial Organs* 2010, **34**(10):862; author reply 862.
40. Bluestein D, Chandran KB, Manning KB: **Towards non-thrombogenic performance of blood recirculating devices.** *Annals Of Biomedical Engineering* 2010, **38**(3):1236-1256.
41. Wootton DM, Markou CP, Hanson SR, Ku DN: **A mechanistic model of acute platelet accumulation in thrombogenic stenoses.** *Annals Of Biomedical Engineering* 2001, **29**(4):321-329.
42. Fogelson AL, Guy RD: **Platelet-wall interactions in continuum models of platelet thrombosis: formulation and numerical solution.** *Mathematical Medicine And Biology: A Journal Of The IMA* 2004, **21**(4):293-334.
43. Leiderman K, Fogelson AL: **Grow with the flow: a spatial-temporal model of platelet deposition and blood coagulation under flow.** *Math Med Biol* 2011, **28**(1):47-84.
44. Xu ZL, Chen N, Kamocka MM, Rosen ED, Alber M: **A multiscale model of thrombus development.** *J R Soc Interface* 2008, **5**(24):705-722.
45. Filipovic N, Kojic M, Tsuda A: **Modelling thrombosis using dissipative particle dynamics method.** *Philos Trans R Soc A-Math Phys Eng Sci* 2008, **366**(1879):3265-3279.
46. Affeld K, Reininger AJ, Gadischke J, Grunert K, Schmidt S, Thiele F: **Fluid mechanics of the stagnation point flow chamber and its platelet deposition.** *Artificial Organs* 1995, **19**(7):597-602.
47. Nyilas E, Morton WA, Cumming RD, Lederman DM, Chiu TH: **Effects of polymer surface molecular structure and force-field characteristics on blood interfacial phenomena.** *Journal Of Biomedical Materials Research* 1977, **11**(1):51-68.
48. Nyilas E, Morton WA, Lederman DM, Chiu TH, Cumming RD: **Interdependence of hemodynamic and surface parameters in thrombosis.** *Transactions - American Society For Artificial Internal Organs* 1975, **21**:55-70.

49. Petschek H, Adamis D, Kantrowitz AR: **Stagnation flow thrombus formation.**
Transactions - American Society For Artificial Internal Organs 1968, **14**:256-260.
50. Solen K, Sukavaneshvar S, Zheng Y, Hanrahan B, Hall M, Goodman P, Goodman B, Mohammad F: **Light-scattering instrument to detect thromboemboli in blood.**
Journal Of Biomedical Optics 2003, **8**(1):70-79.

APPENDIX A FORTRAN ROUTINES

Relevant Fortran subroutines are included here for the reader's convenience. Subroutines provide ample comments to explain the calculations involved in the thrombosis and embolization processes as well as the mechanics behind the parallel processing algorithm.

A.1 Sorsca.f

```
C*****
      SUBROUTINE SORSCA(S1P,S2P)
C      Source-term for scalar species
C*****
C-----*
C      STAR VERSION 4.12.033
C-----*
C
      INCLUDE 'comdb.inc'

      COMMON/USR001/INTFLG(100)

      INCLUDE 'usrdat.inc'
      DIMENSION SCALAR(50)
      EQUIVALENCE( UDAT12(001), ICTID )
      EQUIVALENCE( UDAT03(001), CON )
      EQUIVALENCE( UDAT03(002), TAU )
      EQUIVALENCE( UDAT03(009), DUDX )
      EQUIVALENCE( UDAT03(010), DVDX )
      EQUIVALENCE( UDAT03(011), DWDX )
      EQUIVALENCE( UDAT03(012), DUDY )
      EQUIVALENCE( UDAT03(013), DVDY )
      EQUIVALENCE( UDAT03(014), DWDY )
      EQUIVALENCE( UDAT03(015), DUDZ )
      EQUIVALENCE( UDAT03(016), DVDZ )
      EQUIVALENCE( UDAT03(017), DWDZ )
      EQUIVALENCE( UDAT03(019), VOLP )
      EQUIVALENCE( UDAT04(001), CP )
      EQUIVALENCE( UDAT04(002), DEN )
      EQUIVALENCE( UDAT04(003), ED )
      EQUIVALENCE( UDAT04(004), HP )
      EQUIVALENCE( UDAT04(006), P )
      EQUIVALENCE( UDAT04(008), TE )
      EQUIVALENCE( UDAT04(009), SCALAR(01) )
      EQUIVALENCE( UDAT04(010), SCALAR(02) )
      EQUIVALENCE( UDAT04(011), SCALAR(03) )
```

```

EQUIVALENCE ( UDAT04 (012), SCALAR (04) )
EQUIVALENCE ( UDAT04 (013), SCALAR (05) )
EQUIVALENCE ( UDAT04 (014), SCALAR (06) )
EQUIVALENCE ( UDAT04 (015), SCALAR (07) )
EQUIVALENCE ( UDAT04 (016), SCALAR (08) )
EQUIVALENCE ( UDAT04 (017), SCALAR (09) )
EQUIVALENCE ( UDAT04 (018), SCALAR (10) )
EQUIVALENCE ( UDAT04 (019), SCALAR (11) )
EQUIVALENCE ( UDAT04 (020), SCALAR (12) )
EQUIVALENCE ( UDAT04 (021), SCALAR (13) )
EQUIVALENCE ( UDAT04 (022), SCALAR (14) )
EQUIVALENCE ( UDAT04 (023), SCALAR (15) )
EQUIVALENCE ( UDAT04 (024), SCALAR (16) )
EQUIVALENCE ( UDAT04 (025), SCALAR (17) )
EQUIVALENCE ( UDAT04 (026), SCALAR (18) )
EQUIVALENCE ( UDAT04 (027), SCALAR (19) )
EQUIVALENCE ( UDAT04 (028), SCALAR (20) )
EQUIVALENCE ( UDAT04 (029), SCALAR (21) )
EQUIVALENCE ( UDAT04 (030), SCALAR (22) )
EQUIVALENCE ( UDAT04 (059), U )
EQUIVALENCE ( UDAT04 (060), V )
EQUIVALENCE ( UDAT04 (061), W )
EQUIVALENCE ( UDAT04 (062), VISM )
EQUIVALENCE ( UDAT04 (063), VIST )
EQUIVALENCE ( UDAT04 (007), T )
EQUIVALENCE ( UDAT04 (067), X )
EQUIVALENCE ( UDAT04 (068), Y )
EQUIVALENCE ( UDAT04 (069), Z )
EQUIVALENCE ( UDAT09 (001), IS )

```

```

C-----
C   This subroutine enables the user to specify source terms (per unit
C   volume) for species in linearized form:
C
C   Source = S1P-S2P*SCALAR(IS), (kg/sm3)
C
C   in an arbitrary manner, where S1P and S2P are returned to STAR
C
C   If the species is to be fixed to a given value SCI, then the
C   following may be used:
C   S1P=GREAT*SCI
C   S2P=GREAT
C*****
C   You MUST use the GREAT term in these equations for it to work!!
C*****
C   where SCI can be a constant or an arbitrary function of the
C   parameters in the parameter list.
C-----
C   Sample coding 1: Fix the mass concentration of scalar 1 in fluid 2
C   to a constant value SCALAR(1)=.75
C   IF(IS.EQ.1.AND.IMAT.EQ.2) THEN
C   S1P=GREAT*.75
C   S2P=GREAT
C   ENDIF
C

```

```

C*****
C  Documentation Section
C*****
C  DATE          :: 2012-03-21
C  AUTHOR        :: "Brandon Andersen" <btand@byu.edu>, Denzil Frost
C  -             :: Jeff Ashton, Daniel Lattin, Paul Goodman
C  VERSION       :: 5.0
C  DESCRIPTION   :: Calculate source terms for all species in all cells
C  OPTIONS       :: none

C-----
C  Dimension Variables to be used
C-----
C  Variable types are not explicitly defined
C-----
C  Redefine the various scalars with helpful names:
C-----
      yPl = SCALAR(01)
C  -      platelet mass fraction
      yaPl = SCALAR(02)
C  -      activated platelet mass fraction
      yADP = SCALAR(03)
C  -      ADP mass fraction
      yTxA2 = SCALAR(04)
C  -      Thromboxane A2 mass fraction
      ypT = SCALAR(05)
C  -      proThrombin mass fraction
      yT = SCALAR(06)
C  -      Thrombin mass fraction
      yaT = SCALAR(07)
C  -      antiThrombin III mass fraction
      nBldSt = SCALAR(08)
C  -      blood status of the cell (see key above)
C  -      0 - interior fluid
C  -      1 - wall fluid
C  -      2 - wall thrombus
C  -      3 - interior thrombus
C  -      4 - interior fluid with thrombus neighbor
C  -      5 - wall fluid with thrombus neighbor
C  -      6 - blood vessel wall fluid
      ThrFr = SCALAR(09)
C  -      Thrombus fraction (1=thrombus,0=fluid)
      actFr = SCALAR(10)
C  -      (s) - Time for activation by shear stress
      ShrRt = SCALAR(11)
C  -      (1/s) Shear Rate, calculated in scalfn.f
      xM = SCALAR(15)
C  -      (Pl/m2) platelet concentration on surface
      xMact = SCALAR(16)
C  -      (Pl/m2) activated platelet concentration on surface
      Xact = SCALAR(17)
C  -      fraction of active platelets
      Sp = SCALAR(18)
C  -      fraction of surface not cover by platelets
      PlSrc = SCALAR(19)
C  -      (kg/m3-s) platelet source term (+ to fluid)
      aPlSrc = SCALAR(20)

```

```

C -      (kg/m3-s) activated platelet source term (+ to fluid)
pulse = SCALAR(21)
C -      (Pl/m2) platelets activated at actFr = 1
nPlsFl = SCALAR(22)
C -      activation flag (1=reached on this/prev)

-----
C Define Constants
-----
      wADP = 1
C -      weighting factor for ADP in chemical activation
      wTxA2 = 1
C -      weighting factor for TxA2 in chemical activation
      wT = 1
C -      weighting factor for Thrombin in chemical activation
      cADP = 2E-3
C -      (mol/m3) critical concentration of ADP
      ADPmw = .424
C -      (kg/mol) molecular weight of ADP
      cTxA2 = 6E-4
C -      (mol/m3) critical concentration of TxA2
      TxA2mw = 0.353
C -      (kg/mol) molecular weight of TxA2
      cT = 3.7E-6
C -      (kg/m3) critical concentration of Thrombin
      BldDens = 1056
C -      (kg/m3) - density of blood
      PltMass = 2.12E-15
C -      (kg/platelet) - mass of single blood platelet
      tact = 1.0
C -      (s) characteristic time constant for activation
      rlambda = 1.02E-17
C -      (kg ADP/plt) ADP released by activation of platelet
C -      from 2.4E-8 nmol ADP/platelet (pg.63 of D)
C -      and MW ADP = 0.4244 kg ADP/gmol
C --DT
C -      (s) time step size
      st = 3.35E-21
C -      (kg TxA2/aPlt-s) rate of production of TxA2
C -      from 9.5E-12 nmol TxA2/platelet-s (pg.61 of D)
C -      and MW TxA2 = 0.3525 kg TxA2/gmol
      rki = 0.0161
C -      (1/s) rate constant for hydrolyxation of TxA2 to TxB2
C -      see pg.61 of D
      PltVol = 1.02e-17
C -      (m3/platelet) - volume of a sigle blood platelet
C -      (assumed equal for active and resting platelets)
      xThVd = 0.2
C -      ASSUMED Void fraction in a thrombus
      PHIat = 3.36E-17
C -      (m3/aplt*s)
C -      3.69*10^-9 units/plt*s*microM pT) (pg.64 of D)
C -      *(9.11*10^-6 micromol pT/unit) (1 m3/1000*L)
C -      = 3.36*10^-17 m3/plt*s
      PHIrt = 5.92E-18
C -      (m3/rplt*s)
C -      PHIrt = 6.5*10^-10 units/plt*s*microM pT) (pg.64 of D)

```

```

C -          *(9.11*10^-6 micromol pT/unit) (1 m3/1000*L)
C -          = 5.92*10^-18 m3/plt*s
  xTpT = .508
C -          (kg_T/mol_T) / (kg_pT/mol_pT)
  rk1t = 13.33
C -          (1/s) first order rate constant
  rKat = 1.0E-04
C -          (mol/m3) dissociation constant for heparin/T complex
  rKt = 3.5E-05
C -          (mol/m3) dissociation constant for heparin/aT complex
  H = .5*1.388*(10**4)
C -          (mol/m3) concentration of heparin
C -          this is equal to .5 units/mL
  aTmw = 58.2
C -          (kg/mol) molecular weight of antiThrombin III
  Tmw = 36.0
C -          (kg/mol) molecular weight of Thrombin

C-----
C  Files for output
C-----
  infoFile = 61

C  Tracking File
  myIP = 0
  trkFile = infoFile
C  trkFile = 89
C  open(unit=trkFile, file="mySorsca.txt")
  IF(myIP.EQ.IP) THEN
C  IF(IS.EQ.1) THEN
C  IF(ITER.EQ.1) THEN
    write(trkFile,410) 'LB', 'TIME', 'yPl', 'yaPl', 'yADP', 'yTxA2'
  $      , 'ypT', 'yT', 'yaT'
410  format(A2,A6,A11,A11,A11,A11,A11,A11,A11)
C  ENDIF
    write(trkFile,420) 'SR', TIME, yPl, yaPl, yADP, yTxA2, ypT, yT, yaT
420  format(A2,F6.2,D11.4,D11.4,D11.4,D11.4,D11.4,D11.4,D11.4)
C  ENDIF
  ENDIF

C-----
C  Source Term Calculation
C-----
C  Initialize the source terms (S1P & S2P) to 0.0
  S1P = 0.0
  S2P = 0.0

C-----
C  Agonist-Induced Platelet Activation  pgs. 59-60 of Dissertation
C-----
  IF(myIP.EQ.IP) write(trkFile,*) '-----'
  IF(myIP.EQ.IP) write(trkFile,*) 'IS = ', IS
C  omega = sum (w.j *(a.j/a.j, crit))
  omega = BldDens*(yADP/cADP/ADPmw + yTxA2/cTxA2/TxA2mw + yT/cT)
  IF(myIP.EQ.IP) write(trkFile,*) 'omega = ', omega
  IF (omega.GE.1) THEN
C  if omega < 1.0, kpa (1/s) = 0 (called ratec here)

```

```

    ratec = omega/tact
    ELSE
C   else kpa (1/s) = omega/t.act (called ratec here)
    ratec = 0
    ENDIF
    IF(myIP.EQ.IP) write(trkFile,*) 'ratec = ',ratec
C   Pl_src (kg/m3-s)
    Pl_src = ratec*yPl*BldDens
    IF(myIP.EQ.IP) write(trkFile,*) 'Pl_src = ',Pl_src

C-----
C   Unactivated Platelets
C-----
    IF (IS.EQ.1) THEN
C   Plt (kg_rPlt/m3-s)
    IF (nBldSt.EQ.2.OR.nBldSt.EQ.3) THEN
C   No change in fluid concentration of resting platelets in thrombus
    S1P = 0.0
    ELSE
C   Resting platelet conc change in fluid by chemical activation and
C   transport to the wall
    S1P = -Pl_src + PlSrc
    ENDIF

C-----
C   Activated Platelets
C-----
    ELSEIF (IS.EQ.2) THEN
C   aPlt (kg_aPlt/m3-s)
    IF (nBldSt.EQ.2.OR.nBldSt.EQ.3) THEN
C   No change in fluid concentration of active platelets in thrombus
    S1P = 0.0
    ELSE
C   Active platelet conc change in fluid by chemical activation and
C   transport to the wall
    S1P = Pl_src + aPlSrc
    ENDIF

C-----
C   ADP (see pg. 21 of AC and pg. 59-61 of D) --Same no matter where
C-----
    ELSEIF (IS.EQ.3) THEN
C   ADP (kg_ADp/m3-s)
C   ADP released from agonist-activated platelets
    S1P = rlambda*Pl_src/PltMass
C   ADP released from platelets activated by adhering to wall or
C   thrombus
    S1P = S1P + -PlSrc/PltMass*rlambda
    IF (nPlsFl.EQ.1) THEN
C   ADP released from shear-activated surface platelets (Hellums' Eq.)
    S1P = S1P + rlambda*pulse/VOLP/DT
    ENDIF

C-----
C   Thromboxane A2 (see pg. 21 of AC and pgs. 59,61 of D)
C-----
    ELSEIF (IS.EQ.4) THEN

```



```

C   TxA2 (kg_TxA2/m3-s)
C   TxA2 synthesized in active platelets adheared to surface or wall
      S1P = st*(1-xThVd)*ThrFr*Xact/PltVol
      IF (nBldSt.NE.2.AND.nBldSt.NE.3) THEN
C   TxA2 synthesized in active platelets in bulk fluid
      S1P = S1P + st*yaPl*BldDens/PltMass
      ENDIF
C   TxA2 converted to TxB2 by de-activation
      S1P = S1P - rki*yTxA2*BldDens

-----
C   Prothrombin (see pg. 21 of AC and pgs. 59,61-62,64 of D)
-----
      ELSE IF (IS.EQ.5) THEN
C   pT (kg pT/m3-s)
C   proThrombin consumed in production of Thrombin at platelet surfaces
C   of platelets adhered to platelets or walls
      S1P = -ypT*BldDens*(PHIat*(1-xThVd)*ThrFr*Xact+PHIrt*(1-xThVd)
      $   *ThrFr*(1-Xact))/PltVol
      IF (nBldSt.NE.2.AND.nBldSt.NE.3) THEN
C   proThrombin consumed in the production of Thrombin at platelet
C   surfaces of platelets in bulk fluid
      S1P = S1P - ypT*BldDens*(PHIat*yaPl+PHIrt*yPl)*BldDens/PltMass
      ENDIF

-----
C   Thrombin in Bulk (see pg. 22 of AC and pgs. 59,61-62 of D)
-----
      ELSE IF (IS.EQ.6) THEN
C   tgen (kg_T/m3-s)
C   Generation of thrombin by proThrombin inside thrombus at surface of
C   platelets
      tgen = ypT*BldDens*(PHIat*(1-xThVd)*ThrFr*Xact+PHIrt*(1-xThVd)
      $   *ThrFr*(1-Xact))*xTpT/PltVol
C   tloss (kg_T/m3-s) = Gamma * [T]
C   heparin-catalyzed inactivation of Thrombin by antiThrombin III in
C   bulk fluid and thrombus
      tloss = rklt*H*yaT*BldDens*yT*BldDens/(aTmw*(rKat*rKt + rKat*yT
      $   *BldDens/Tmw + rKt*yaT*BldDens/aTmw + yaT*yT*BldDens*BldDens
      $   / (aTmw*Tmw)))
      IF (nBldSt.NE.2.AND.nBldSt.NE.3) THEN
C   generation of thrombin by prothrombin at platelet surfaces in bulk
C   fluid
      tgen = tgen + ypT*BldDens*(PHIat*yaPl+PHIrt*yPl)*BldDens*xTpT
      $   /PltMass
      ENDIF
C   T (kg_T/m3-s) = tgen - tloss
      S1P = tgen - tloss

-----
C   antiThrombin III (see pg. 22 of AC and pgs. 59,61-64 of D)
-----
      ELSE IF (IS.EQ.7) THEN
C   aT (kg_aT/m3-s) = -Gamma*[T]*aTmw/Tmw
C   Consumption of antiThrombin III to inactivate Thrombin in bulk fluid
C   and thrombus
      S1P = -rklt*H*yaT*BldDens*yT*BldDens/(aTmw*(rKat*rKt + rKat*yT

```

```

$      *BldDens/Tmw + rKt*yaT*BldDens/aTmw + yaT*yT*BldDens*BldDens
$      / (aTmw*Tmw)) *aTmw/Tmw
ENDIF

IF(myIP.EQ.IP) write(trkFile,*) 'S1P = ',S1P
IF(myIP.EQ.IP) write(trkFile,*) 'S2P = ',S2P

RETURN
END

```

A.2 Posdat.f

```

C*****
C      subroutine posdat(level)
C      Post-process data
C*****
C-----
C      STAR-CD VERSION 4.16.000
C-----
C      USE allmod
C      USE aaaUsrMod
C      USE timeParallel, ONLY: globalMaster
C      IMPLICIT NONE
C      INCLUDE 'std.inc'
C-----
C
C      This subroutine enables the user to output data and is called
C      at the beginning and at the end of each iteration/time step,
C      i.e.
C      if (level.eq.1) then
C.>      called at the beginning of iteration/time step
C      else if (level.eq.2) then
C.>      called at the end of iteration/time step
C      end if
C      Any user code which is not enclosed in the IF condition will
C      be executed for both calls
C
C      Note: 1. File units available to the users for opening their own
C             files are from 84 to 89. Users may write to unit 6 or 60
C             if they want to see their output on the terminal or
C             the run file.
C           2. All variables passed to this routine use STAR cell
C             numbering, which is different from pro-STAR cell numbers.
C             pro-STAR cell number can be obtained from a STAR cell
C             number ICSTAR by ICPROSTAR=ICLMAP(ICSTAR)
C-----
C*****
C      Documentation Section
C*****
C      DATE      :: 2012-03-21
C      AUTHOR    :: "Brandon Andersen" <btand@byu.edu>, Denzil Frost
C      -        :: Jeff Ashton, Daniel Lattin, Paul Goodman
C      VERSION  :: 5.0

```

```

C DESCRIPTION :: calculate platelet adhesion to walls and thrombi,
C -           :: manage thrombosis and embolization
C OPTIONS     :: none

C-----
C Functions/Calls Used
C-----
C ### regi(nr)%type
C Returns the type of region nr

C ### Call FsetBoun(fs,0,ND_ALL,NSD_ALL,nr,ISIDE_ALL)
C Creates a set of faces (fs) that meet some criteria

C ### lfs(1,IS)
C Identifies the cell on the inside(1) of the boundary face IS

C ### Call cset(cs,0,nd,NSD_ALL,INTERNAL)
C Returns a set of cells (cs) that meet some criteria

C ### call arrSwap(var,nd,index,icomp,FSFLAG)
C provided by Senthil Dhanapalan at CD Adapco
C Swaps the values of the selected variables between the cells on the
C boundary of each processor domain with the cells just beyond that
C boundary (halo cells for that processor)
C var(*),var(*,*):integer,single/double precision array to be swapped
C nd          : swap data of domain nd
C index       : element (1, 2, 3...) in the array list that is to be fixed
C icomp       : component to be fixed
C              e.g. index=1, icomp=2 => send var(2,:),
C              or index=2, icomp=7 => send var(:,7)
C FSFLAG      : Allowed values are FLUID,SOLID,FLSOL (1,2,3)
C              swap only if the domain is F,S or both

C ### GSUM(localVariable) --see star_uguide.pdf Appendix D
C Returns the real/double sum of 'localVariable' from all processors

C ### IGSUM(localVariable) --see star_uguide.pdf Appendix D
C Returns the integer sum of 'localVariable' from all processors

C ### Call LGLOR(locLogical,glblLogical) --see star_uguide.pdf
C Appendix D
C Returns the OR result of comparing all 'locLogical' and stores it
C as 'glblLogical'

C-----
C User Defined Arrays
C-----
C :: ThrArray(row) :: is a global array (updated on all processors)
C in which the IDs (ThrNum) of all active thrombi are stored

C :: ThrLoc(row) :: is a local array (populated separately by each
C processor and cleared after each time step) in which the IDs
C (iThrNum) of new thrombi within that processor are stored

C :: ThrMap(iThrNum) :: is a global array (updated on all processors)
C that returns the row of ThrArray in which iThrNum is stored

```

```

C  :: ThrLg(nT)  :: is a logical that tells if any INTERNAL cell
C  in the processor belongs to the thrombus 'nT'

C  :: ThrLgH(nT)  :: is a logical that tells if any HALO cell for
C  the processor belongs to the thrombus 'nT'

C  :: ThrMP(nT)  :: is a logical array that identifies a thrombus
C  that exists on multiple processors (only one processor
C  will correctly identify the thrombus as .TRUE.)

C  :: ThrSA(nT)  :: is Real array that temporarily stores the area
C  of some thrombus that touches a neighboring thrombus nT

C  :: c_old(ic)  :: is a real array that stores calculation
C  temporarily to compare to what is returned from the arrSwap

C  :: c_old2(ic)  :: is a real array that stores calculation
C  temporarily to compare to what is returned from the arrSwap

C  :: NeArray(row)  :: is a global array (updated on all processors)
C  in which the IDs (ThrNum) of all thrombi which are neighbors to the
C  thrombus being analyzed are temporarily stored

C*****
C  Code Section
C*****
C-----
C  Dimension REAL and INTEGER variables to be used
C-----
      REAL(ra)  actT,AdhStrs,apadhm,athromb
      REAL(ra)  BldDens,BldStSet,BldVisc,CoalArea,cVol,delta,deltau
      REAL(ra)  frpremain,height,kas,kt,kus,MAX,myTime
      REAL(ra)  nbraPlt,nbrPlt,nbrXact,percCA,PltMass,PltVol
      REAL(ra)  SA1,SAsum,SAthromb, shearfc, sheartot, surfarea
      REAL(ra)  Threshold,upadhm
      REAL(ra)  uthromb,vthromb,vtot,xMinf,xThVd

      INTEGER iactFr,iaPlSrc,IC,iEmbVol,iIHPC,iNumEmb,inBldSt
      INTEGER iArBnd,iArThr,inPlsFl,iPlSrc,iPLTadh,ipulse
      INTEGER IS,iShrRt,iSp,iThrNum,iThrFr,iFrTmThr
      INTEGER iviscosity,ixM,ixMact,iXact,iyPl,iyaPl,level
      INTEGER MAXNBR,NBRCLL,ncount,nd,nf,nfs,nr,nset,nT,nT1
      INTEGER Thrneighbor

      INTEGER allocatestatus,GNCELL,itMax,itNum
      INTEGER infoFile,embFile,errFile,trkFile
      INTEGER intMax,intMin,it,MAXROW
      INTEGER myCell,myICLMAP,myRow,myThromb,myVar,numZero
      INTEGER row,row1,row2,vThrFile
      LOGICAL CmbThr,CmbThrG,EmbThr,EmbThrG,errDiag,errFnd
      LOGICAL needSwap,needSwapG,ThrMPG

C-----
C  Cell property storage location in array c(cell,properties)
C-----
      iyPl = 1
C  -      platelet mass fraction

```

```

    iyaPl = 2
C -      activated platelet mass fraction
    inBldSt = 8
C -      blood status of the cell (see key above)
C -      0 - interior fluid
C -      1 - wall fluid
C -      2 - wall thrombus
C -      3 - interior thrombus
C -      4 - interior fluid with thrombus neighbor
C -      5 - wall fluid with thrombus neighbor
C -      6 - blood vessel wall fluid
C -      7 - flow outlet (assigned temporarily)
    iThrFr = 9
C -      Thrombus fraction (1=thrombus,0=fluid)
    iactFr = 10
C -      Fractional progression toward shear activation
    iShrRt = 11
C -      (1/s) Shear Rate, calculated in scalfn.f
    iIHPC = 12
C -      stores the processor number to which the cell belongs
C -      or a zero if the cell is on a boundary between processors
    iArBnd = 13
C -      (m2) Contact area of the cell against a boundary
    iArThr = 14
C -      (m2) Contact area of the cell against a thrombus
    ixM = 15
C -      (Pl/m2) Platelet concentration on surface
    ixMact = 16
C -      (Pl/m2) Activated platelet concentration on surface
    iXact = 17
C -      Fraction of active platelets
    iSp = 18
C -      Fraction of surface not cover by platelets
    iPlSrc = 19
C -      (kg/m3-s) Platelet source term (+ to fluid)
    iaPlSrc = 20
C -      (kg/m3-s) Activated platelet source term (+ to fluid)
    ipulse = 21
C -      (Pl) Platelets activated upon shear activation
    inPlsFl = 22
C -      Shear Activation flag (1=reached on this/prev)
    iPLTadh = 23
C -      (Pl/s) Platelet adhesion rate (+ to wall)
    iThrNum = 24
C -      Thrombus to which this cell belongs
    iNumEmb = 25
C -      Total number of times the cell has embolized
    iFrTmThr = 26
C -      Fraction of time the cell has been thrombus
    iEmbVol = 27
C -      Average volume of emboli from this cell
    iviscosity = 28
C -      (kg/m-s,poise) Viscosity of cell

```

```

-----
C   Program Constants
-----

```

```

      BldDens = 1056
C   -          (kg/m3) - Density of blood
      BldVisc = 0.0035
C   -          (Pa*s) - Viscosity of blood
      kas = 0.0035
C   -          (m/s) - Rate constant, activated platelets to wall
      kt = 3.5E-3
C   -          (m/s) - Rate constant, platelets to adhered platelets
      kus = 0.0000025
C   -          (m/s) - Rate constant, unactivated platelets to wall
      PltMass = 2.12E-15
C   -          (kg/platelet) - Mass of single blood platelet
      PltVol = 1.02e-17
C   -          (m3/platelet) - Volume of a single blood platelet
      Pi = 3.1415926
C   -          (radians) Value of pi
      xMinf = 1.132E11
C   -          (Pl/m2) Maximum possible platelet surface coverage
      xThVd = 0.2
C   -          ASSUMED Void fraction in a thrombus
      AdhStrs = 6.5000
C   -          (N/m2,Pa) - Adhesion expressed as Stress
      CoalArea = 0.10
C   -          Fractional Thr SA that must be shared for coalescence
      itMax = 10
C   -          Maximum number of allowed iterations for swap
      GNCELL = IGSUM(NCELL)
C   -          Total number of cells in the model

C-----
C   Files for output
C-----
      infoFile = 61

C---Embolism File
      embFile = infoFile
C   embFile = 86
C   open(unit=embFile, file = "myEmboli.txt", position='append')

C---Total Thrombus Volume File
      vThrFile = 87
      open(unit=vThrFile, file = 'myThromb.txt', position='append')

C-----
C   CODE EXECUTED
C-----
      IF(level.EQ.1) THEN

C#####
C   DO THE FOLLOWING IN THE FIRST CALL TO POSDAT (ie BEFORE CALCULATION)

C-----
C   DO THE FOLLOWING AT THE BEGINNING OF THE FIRST TIME STEP
      IF(ITER.EQ.1) THEN

C   Allocate memory for the user defined arrays
      ALLOCATE(ThrLg(1:GNCELL),stat=allocatestatus)

```

```

if (allocatestatus.ne.0) stop
$   '*** not enough memory for ThrLg() array ***'
Do row=1,GNCELL
  ThrLg(row) = 0
Enddo
ALLOCATE(ThrLgH(1:GNCELL),stat=allocatestatus)
if (allocatestatus.ne.0) stop
$   '*** not enough memory for ThrLgH() array ***'
Do row=1,GNCELL
  ThrLgH(row) = 0
Enddo
ALLOCATE(ThrLoc(1:NCELL),stat=allocatestatus)
if (allocatestatus.ne.0) stop
$   '*** not enough memory for ThrLoc() array ***'
Do row=1,NCELL
  ThrLoc(row) = 0
Enddo
ALLOCATE(ThrMP(1:GNCELL),stat=allocatestatus)
if (allocatestatus.ne.0) stop
$   '*** not enough memory for ThrMP() array ***'
Do row=1,GNCELL
  ThrMP(row) = 0
Enddo
ALLOCATE(ThrMAP(1:GNCELL),stat=allocatestatus)
if (allocatestatus.ne.0) stop
$   '*** not enough memory for ThrMAP() array ***'
Do row=1,GNCELL
  ThrMAP(row) = 0
Enddo
ALLOCATE(ThrArray(1:GNCELL),stat=allocatestatus)
if (allocatestatus.ne.0) stop
$   '*** not enough memory for ThrArray() array ***'
Do row=1,GNCELL
  ThrArray(row) = 0
Enddo
ALLOCATE(ThrSA(1:GNCELL),stat=allocatestatus)
if (allocatestatus.ne.0) stop
$   '*** not enough memory for ThrSA() array ***'
Do row=1,GNCELL
  ThrSA(row) = 0
Enddo
ALLOCATE(c_old(1:GNCELL),stat=allocatestatus)
if (allocatestatus.ne.0) stop
$   '*** not enough memory for c_old() array ***'
Do row=1,GNCELL
  c_old(row) = 0
Enddo
ALLOCATE(c_old2(1:GNCELL),stat=allocatestatus)
if (allocatestatus.ne.0) stop
$   '*** not enough memory for c_old2() array ***'
Do row=1,GNCELL
  c_old2(row) = 0
Enddo
ALLOCATE(NeArray(1:NCELL),stat=allocatestatus)
if (allocatestatus.ne.0) stop
$   '*** not enough memory for NeArray() array ***'
Do row=1,NCELL

```

```

        NeArray(row) = 0
    Enddo

C   Populate the iIHPC value for all cells in the model. IHPC
C   identifies the processor to which the cell has been assigned
    do nd=1,doma_no
        call cset(cs,0,nd,NSD_ALL,INTERNAL)
        do nset=1,cs%no
            do IC=cs%ns(nset),cs%ne(nset)
                c(IC,iIHPC) = IHPC
            enddo
        enddo
    enddo

C   Make the iIHPC value 0 for cells on a processor boundary
    do nd=1,doma_no
        call cset(cs,0,nd,NSD_ALL,HALO)
        do nset=1,cs%no
            do IC=cs%ns(nset),cs%ne(nset)
                do nf=1,c2f(IC)%No
                    IS = c2f(IC)%List(nf)
C   Ignore negative (boundary) faces
                    IF(IS.GT.0) THEN
                        do nfs=1,2
                            IF(lfc(nfs,IS).NE.IC) THEN
                                NBRCLL = LFC(nfs,IS)
                                c(NBRCLL,iIHPC) = 0
                            ENDIF
                        enddo
                    ENDIF
                enddo
            enddo
        enddo
    enddo

C   All boundary cells are identified by finding all boundary faces and
C   then their corresponding cells. The blood status of cells next to a
C   wall (or a blood vessel) is set.
    itNum = 0
10   continue
    itNum = itNum + 1

C   The c_old2 array is used to tell if a cell started the iteration
C   with blood status not set as a wall cell. This is useful after the
C   swap to prevent doubling the area of all cells against the boundary
    do nd=1,doma_no
        call cset(cs,0,nd,NSD_ALL,INT_HALO)
        do nset=1,cs%no
            do IC=cs%ns(nset),cs%ne(nset)
                IF(c(IC,inBldSt).EQ.0.0) then
                    c_old2(IC) = 0.0
                ELSE
                    c_old2(IC) = 1.0
                ENDIF
            enddo
        enddo
    enddo

C   For cells against the boundary, change blood status and calculate
C   area against the boundary

```



```

do nr=0,regi_no
  if(regi(nr)%type.EQ.WALL) then
    if(nr.EQ.1) BldStSet = 1.0
    if(nr.EQ.5) BldStSet = 6.0
    call FsetBoun(fs,0,ND_ALL,NSD_ALL,nr,ISIDE_ALL)
    do nset=1,fs%no
      do IS=fs%ns(nset), fs%ne(nset)
        IC = lfc(1,IS)
        c_old(IC) = BldStSet
        IF(c_old2(IC).EQ.0.0) THEN
          c(IC,inBldSt) = BldStSet
          c(IC,iArBnd) = c(IC,iArBnd) + sqrt(sv(1,IS)**2+sv(2,IS)
$           **2+sv(3,IS)**2)
          ENENDIF
        enddo
      enddo
    endif
  enddo
C Populate the c_old array for halo cells (loop above only populates
C internal cells)
  do nd=1,doma_no
    call cset(cs,0,nd,NSD_ALL,HALO)
    do nset=1,cs%no
      do IC=cs%ns(nset),cs%ne(nset)
        c_old(IC) = c(IC,inBldSt)
      enddo
    enddo
  enddo
C If this is a parallel run, swap/assign repeatedly until the BldSt
C values stop changing
  if (parrun) then
    needSwap = .FALSE.
C Perform swap for Blood Status and area against a boundary
    do nd=1,doma_no
      call arrSwap(c,nd,2,inBldSt,FLUID)
      call arrSwap(c,nd,2,iArBnd,FLUID)
    end do
c Check if the field values have changed in this processor
    do nd=1,doma_no
      call cset(cs,0,nd,NSD_ALL,INT_HALO)
      do nset=1,cs%no
        do IC=cs%ns(nset),cs%ne(nset)
          if (c(IC,inBldSt).ne.c_old(IC).AND.itNum.lt.itMax) then
            needSwap = .TRUE.
            goto 15
          endif
        end do
      end do
    end do
15 continue
C If field values changed on any processor, perform swapp again
    call LGLOR(needSwap,needSwapG)
    if(needSwapG) goto 10
  end if
C Clear out c_old and c_old2 arrays
  DO nd=1,doma_no
    if(doma(nd)%mattyp.eq.FLUID) then

```

```

    call cset(cs,0,nd,NSD_ALL,INT_HALO)
    do nset=1,cs%no
      do IC=cs%ns(nset),cs%ne(nset)
        c_old(IC) = 0
        c_old2(IC) = 0
      enddo
    enddo
  endif
ENDDO

C Exclude outlet region from calculations
do nr=0,regi_no
  if(regi(nr)%type.EQ.2) then
    call FsetBoun(fs,0,ND_ALL,NSD_ALL,nr,ISIDE_ALL)
    do nset=1,fs%no
      do IS=fs%ns(nset), fs%ne(nset)
        IC = lfc(1,IS)
        c(IC,inBldSt) = 7
      enddo
    enddo
  endif
enddo
do it=1,30
  do nd=1,doma_no
    call cset(cs,0,nd,NSD_ALL,INT_HALO)
    do nset=1,cs%no
      do IC=cs%ns(nset),cs%ne(nset)
        if(c(IC,inBldSt).EQ.7) then
          do nf=1,c2f(IC)%No
            IS = c2f(IC)%List(nf)
C Ignore negative (boundary) faces
            IF(IS.GT.0) THEN
              do nfs=1,2
                IF(lfc(nfs,IS).NE.IC) THEN
                  NBRCLL = LFC(nfs,IS)
                  c(NBRCLL,inBldSt) = 7
                ENDIF
              enddo
            ENDIF
          enddo
        endif
      enddo
    enddo
  enddo
enddo

C If this is a parallel run, swap values between processors
  if (parrun) then
C Perform swap for Blood Status
    do nd=1,doma_no
      call arrSwap(c,nd,2,inBldSt,FLUID)
    end do
  end if
enddo

C Replace blood status 7 with blood status 0
do nd=1,doma_no
  call cset(cs,0,nd,NSD_ALL,INT_HALO)
  do nset=1,cs%no
    do IC=cs%ns(nset),cs%ne(nset)

```

```

        if(c(IC,inBldSt).EQ.7) then
            c(IC,inBldSt) = 0
        endif
    enddo
enddo
enddo

C-----
C-----
C DO THE FOLLOWING AT THE BEGINNING OF ALL TIME STEPS (AFTER 1ST)
  ELSE

C Loop through all of the cells
  DO nd=1,doma_no
    if(doma(nd)%mattyp.eq.FLUID) then
      call cset(cs,0,nd,NSD_ALL,INTERNAL)
      do nset=1,cs%no
        do IC=cs%ns(nset),cs%ne(nset)
          delthu = 0.0
          deltha = 0.0
          c(IC,iPlSrc) = 0.0
          c(IC,iaPlSrc) = 0.0
          c(IC,iPLTadh) = 0.0

C Update the fraction of time filled with thrombus value
          if(c(IC,inBldSt).EQ.2.OR.c(IC,inBldSt).EQ.3) then
            c(IC,iFrTmThr) = (c(IC,iFrTmThr)*(TIME-DT)+DT)/TIME
          endif

C Calculate the progression toward time of activation of thrombus. Use
C Hellums' equation (see pp. 73 of dissertation) where 0.035 is the
C viscosity of blood in POISE. This equation requires that the shear
C rate be in units of s^-1
          IF((c(IC,inBldSt).EQ.1.AND.c(IC,iShrRt).LT.20).OR.(c(IC
          $ ,inBldSt).GE.2.AND.c(IC,inBldSt).LE.5)) THEN
            IF(c(IC,iShrRt).GT.0.0) THEN
C Calculate the total time required for activation at the current
C conditions
              actT = 4.0E6*(c(IC,iShrRt)*0.035)**(-2.3)
C Calculate the change in the overall progression toward shear
C activation
              c(IC,iactFr) = c(IC,iactFr) + DT/actT
            ENDIF
          ENDIF

C If shear activation has been achieved
          IF(c(IC,iactFr).GE.1.0) THEN
C If the cell is not already activated
          IF(c(IC,inPlsFl).EQ.0.0) THEN
C Calculate the "pulse" of activated platelets and set pulse flag.
          c(IC,ipulse) = VOL(IC)*c(IC,iThrFr)*(1-c(IC,iXact))*(1
          $ -xThVd)/PltVol
C Change the pulse flag for when activation time has been reached.
          c(IC,inPlsFl) = 1.0
C Set the activated concentration to the total concentration and
C fraction activated to 1.

```

```

        c(IC,ixMact) = c(IC,ixM)
        c(IC,ixAct) = 1.0
C   If the cell was activated in the previous time step, reset values
        ELSEIF (c(IC,inPlsFl).EQ.1.0) THEN
            c(IC,ipulse) = 0.0
            c(IC,inPlsFl) = 0.0
            c(IC,iactFr) = 0.0
        ENDIF
    ENDIF

C   Calculate source terms for Pl, aPl--store as scalars for use in
C   sorsca.f
C   The restriction has been made that for wall fluid to undergo these
C   calculations, the shear rate must be less than 20 1/s This comes
C   from work by Hubbell and McIntire. Visualization and analysis of
C   mural thrombogenesis on collagen, polyurethane and nylon.
C   Biomaterials 1986;7:354-63 (mentioned in Corbett et al.,2010)
        IF((c(IC,inBldSt).EQ.1.0.AND.c(IC,iShrRt).LT.20).OR.c(IC
    $       ,inBldSt).EQ.5.0.OR.c(IC,inBldSt).EQ.4.0) THEN
            uthromb = 0.0
            athromb = 0.0
            upadhm = 0.0
            apadhm = 0.0
            cVol = VOL(IC)

C   Calculate the fraction of platelets that do not embolize in the time
C   step (frpremain). This comes from Goodman's data making the
C   assumption that the removal of platelets under a shear force is
C   instantaneous
C   -3.325e-4(s) = -.0095(cm2/dyn)*.035(poise)
        frpremain = EXP(-3.325e-4*c(IC,iShrRt))

C-----
C   DO THE FOLLOWING CALCULATIONS ONLY FOR WALL FLUID
        IF(c(IC,inBldSt).EQ.1.0.OR.c(IC,inBldSt).EQ.5.0) THEN

C   Calculate the mass rate of resting platelets adhering to the wall
C   upadhm (kg/s) = -c(IC,iSp)*kus*[Pl]*c(IC,iArBnd)
        upadhm = -c(IC,iSp)*kus*c(IC,iyPl)*BldDens*c(IC,iArBnd)
    $       *frpremain

C   Calculate the mass rate of active platelets adhering to the wall
C   apadhm (kg/s) = -c(IC,iSp)*kas*[aPl]*c(IC,iArBnd)
        apadhm = -c(IC,iSp)*kas*c(IC,iyaPl)*BldDens*c(IC,iArBnd)
    $       *frpremain

C   Calculate the rate of unactivated platelets adhering to activated
C   platelets on the surface (i.e. forming uthrombs)
C   uthromb (kg/s) = -Mact/Minf*kt*[aPl]*c(IC,iArBnd)
        uthromb = -c(IC,ixAct)*(1-c(IC,iSp))*c(IC,iArBnd)*kt*c(IC
    $       ,iyPl)*BldDens*frpremain

C   Calculate the rate of activated platelets adhering to activated
C   platelets on the surface
C   athromb (kg/s) = -Mact/Minf*kt*[aPl]*c(IC,iArBnd)
        athromb = -c(IC,ixAct)*(1-c(IC,iSp))*c(IC,iArBnd)*kt*c(IC
    $       ,iyaPl)*BldDens*frpremain

```

```

C Calculate the surface concentration of platelets (Plt/m2)
  c(IC,ixM) = c(IC,ixM) - (upadhm + apadhm)*DT/PltMass
  $
    /c(IC,iArBnd)
    IF(c(IC,ixM).LT.0.0) c(IC,ixM) = 0.0
    IF(c(IC,ixM).GT.xMinf) c(IC,ixM) = xMinf

C Calculate the fraction of boundary area free of platelets
  c(IC,iSp) = 1 - (c(IC,ixM)/xMinf)

C Calculate the surface concentration of active platelets (Plt/m2)
  c(IC,ixMact) = c(IC,ixMact) - (apadhm)*DT/PltMass
  $
    /c(IC,iArBnd)
    IF(c(IC,ixMact).GT.c(IC,ixM)) c(IC,ixMact) = c(IC,ixM)
    IF(c(IC,ixMact).LT.0) c(IC,ixMact) = 0.0

    ENDIF

C-----
C DO THESE CALCULATIONS ONLY FOR FLUID WITH THROMBUS NEIGHBOR
  IF(c(IC,inBldSt).EQ.4.0.OR.c(IC,inBldSt).EQ.5.0) THEN
C If the thrombus is new (so that we don't know anything about the
C relative number of active vs resting platelets to which platelets
C may adhere) then find the Xact of the neighboring thrombus cells.
  IF(abs(c(IC,iXact)).LT.1E-16) THEN
    nbrPlt = 0
    nbraPlt = 0
    nbrXact = 0
    do nf=1,c2f(IC)%No
      IS = c2f(IC)%List(nf)
C Ignore negative (boundary) faces
      IF(IS.GT.0) THEN
        do nfs=1,2
          IF(lfc(nfs,IS).NE.IC) THEN
            NBRCLL = LFC(nfs,IS)
            IF(c(NBRCLL,iThrNum).GT.1E-4) THEN
              nbrPlt = nbrPlt + VOL(NBRCLL)*(1-xThVd)
              nbraPlt = nbraPlt + VOL(NBRCLL)*(1-xThVd)*c(NBRCLL
              $
                ,iXact)
            ENDIF
          ENDIF
        enddo
      ENDIF
    enddo
    nbrXact = nbraPlt/nbrPlt
C Calculate the rate of resting platelets adhering to platelets on
C the thrombus
C uthromb (kg/s) = uthromb - kt*[Pl]*c(IC,iArThr)
    uthromb = uthromb - kt*c(IC,iyPl)*BldDens*c(IC,iArThr)
    $
      *nbrXact*frpremain
C Calculate the rate of active platelets adhering to platelets on
C the thrombus
C athromb (kg/s)= athromb - kt*[aPl]*iArThr
    athromb = athromb - kt*c(IC,iyaPl)*BldDens*c(IC,iArThr)
    $
      *nbrXact*frpremain

    ELSE

```

```

C   In the case that we know the fraction of adhered platelets in the
C   cell that are active, scale the available surface area
C   appropriately
C   Calculate the rate of resting platelets adhering to platelets on
C   the thrombus
C   uthromb (kg/s) = uthromb - kt*[Pl]*c(IC,iArThr)
C       uthromb = uthromb - kt*c(IC,iyPl)*BldDens*c(IC,iArThr)
C   $
C       *c(IC,iXact)*frpremain
C   Calculate the rate of active platelets adhering to platelets on
C   the thrombus
C   athromb (kg/s)= athromb - kt*[aPl]*iArThr
C       athromb = athromb - kt*c(IC,iyaPl)*BldDens*c(IC,iArThr)
C   $
C       *c(IC,iXact)*frpremain
C       ENDIF
C   ENDIF

C-----
C   Calculate the change in volumetric fraction /s of the cell occupied
C   by uthrombs and athrombs
C       delthu = -(uthromb+upadhm)*PltVol/cVol/PltMass/(1-xThVd)
C       deltha = -(athromb+apadhm)*PltVol/cVol/PltMass/(1-xThVd)

C   Calculate the fraction of adhered platelets in the cell that are
C   active
C   Xact (aPlt/Plt)
C       IF(abs(c(IC,iThrFr)+(deltha+delthu)*DT).LT.1E-20) THEN
C           c(IC,iXact) = 0
C       ELSE
C           c(IC,iXact) = (c(IC,iXact)*c(IC,iThrFr) + deltha*DT)
C   $
C           / (c(IC,iThrFr) + (deltha+delthu)*DT)
C       ENDIF

C   Calculate the new thrombus fraction
C       c(IC,iThrFr) = c(IC,iThrFr) + (deltha + delthu)*DT
C       IF(c(IC,iThrFr).GT.0.99) c(IC,iThrFr) = 1.0
C       IF(c(IC,iThrFr).LT.0.0) c(IC,iThrFr) = 0.0

C   Update the source terms
C       c(IC,iPlSrc) = (upadhm+uthromb)/cVol
C       c(IC,iaPlSrc) = (apadhm+athromb)/cVol
C       c(IC,iPlTadh) = -(upadhm+apadhm)/PltMass

C       ENDIF
C   enddo
C   enddo
C   endif
C   ENDDO

C   Now that the computations are complete, we focus on multi-processor
C   communication and assignments for new thrombus cells
C   Update the blood status for new thrombus cells & communicate across
C   processors
C       itNum = 0
C   20   continue
C       itNum = itNum + 1
C   Loop through all of the cells
C       DO nd=1,doma_no

```

```

        if(doma(nd)%mattyp.eq.FLUID) then
            call cset(cs,0,nd,NSD_ALL,INTERNAL)
            do nset=1,cs%no
                do IC=cs%ns(nset),cs%ne(nset)
C      Look at all cells that are thrombus
                    IF(c(IC,iThrFr).GT.0.99) THEN
C      Change blood status of cells that became thrombus
                        IF(c(IC,inBldSt).EQ.0.0.OR.c(IC,inBldSt).EQ.4.0) THEN
                            c(IC,inBldSt)=3.0
                        ELSE IF(c(IC,inBldSt).EQ.1.0.OR.c(IC,inBldSt).EQ.5.0) THEN
                            c(IC,inBldSt)=2.0
                        ENDIF
                    ENDIF
                    c_old(IC) = c(IC,inBldSt)
                enddo
            enddo
        endif
    ENDDO

    do nd=1,doma_no
        if(doma(nd)%mattyp.eq.FLUID) then
            call cset(cs,0,nd,NSD_ALL,HALO)
            do nset=1,cs%no
                do IC=cs%ns(nset),cs%ne(nset)
                    c_old(IC) = c(IC,inBldSt)
                enddo
            enddo
        endif
    enddo

c      If this is a parallel run, swap/assign repeatedly until the values
c      stop changing
        if (parrun) then
            needSwap = .FALSE.
            do nd=1,doma_no
                call arrSwap(c,nd,2,inBldSt,FLUID)
            end do
            do nd=1,doma_no
                call cset(cs,0,nd,NSD_ALL,INT_HALO)
                do nset=1,cs%no
                    do IC=cs%ns(nset),cs%ne(nset)
c      If the field values have changed, repeat the swap
                        if (c(IC,inBldSt).ne.c_old(IC).AND.itNum.lt.itMax) then
                            needSwap = .TRUE.
                            goto 21
                        endif
                    end do
                end do
            end do
        end if
21      continue
        call LGLOR(needSwap,needSwapG)
        if(needSwapG) goto 20
C      Clear out c_old array
        DO nd=1,doma_no
            if(doma(nd)%mattyp.eq.FLUID) then

```

```

        call cset(cs,0,nd,NSD_ALL,INT_HALO)
        do nset=1,cs%no
            do IC=cs%ns(nset),cs%ne(nset)
                c_old(IC) = 0
            enddo
        enddo
    endif
ENDDO

C    Update viscosity,fraction active, thrombus neighbor BldSt
C    Loop through all of the cells
        DO nd=1,doma_no
            if(doma(nd)%mattyp.eq.FLUID) then
                call cset(cs,0,nd,NSD_ALL,INT_HALO)
                do nset=1,cs%no
                    do IC=cs%ns(nset),cs%ne(nset)
C    Identify thrombus cells
                        IF(c(IC,inBldSt).EQ.2.OR.c(IC,inBldSt).EQ.3) THEN
C***Increase the viscosity by 10%. stop at 1 poise
                            IF(c(IC,iViscosity).GE.0.9) THEN
                                c(IC,iViscosity) = 1
                            ELSE if(c(IC,iViscosity).LT.0.9) THEN
                                c(IC,iViscosity)=c(IC,iViscosity)*1.1
                            ENDIF
C***Increase the fraction of active platelets in thrombus cells at a
C    constant rate to achieve a fraction of one
                            IF(c(IC,iXact).LT.1) THEN
                                c(IC,iXact) = c(IC,iXact) + 1/60
                            ELSE
                                c(IC,iXact) = 1
                            ENDIF
C    Update blood status of neighboring cells
                                do nf=1,c2f(IC)%No
                                    IS = c2f(IC)%List(nf)
C    Ignore negative (boundary) faces
                                    IF(IS.GT.0) THEN
                                        do nfs=1,2
                                            IF(lfc(nfs,IS).NE.IC) THEN
                                                NBRCLL = LFC(nfs,IS)
C    --- If the neighbor cell is fluid, change its nBldSt
                                                IF(c(NBRCLL,inBldSt).EQ.0.0) THEN
                                                    c(NBRCLL,inBldSt) = 4.0
                                                ELSE IF(c(NBRCLL,inBldSt).EQ.1.0) THEN
                                                    c(NBRCLL,inBldSt) = 5.0
                                                ENDIF
                                                c_old(NBRCLL) = c(NBRCLL,inBldSt)
                                            ENDIF
                                        enddo
                                    ENDIF
                                enddo
C    Identify non-thrombus cells
                                ELSE
C***Lower the viscosity of cells where a thrombus recently embolized.
                                    IF(c(IC,iViscosity).GE.0.07) THEN
                                        c(IC,iViscosity) = c(IC,iViscosity)/20
                                    ELSE IF(c(IC,iViscosity).LT.0.07) THEN
                                        c(IC,iViscosity) = BldVisc
                                    ENDIF
                                ENDIF
                            ENDIF
                        enddo
                    enddo
                enddo
            enddo
        END DO

```



```

        ENDIF
    ENDIF
    enddo
enddo
endif
ENDDO

C   Communicate iXact accross processors
C   Internal cells values are accurate, record them in c_old2
    DO nd=1,doma_no
        if (doma(nd)%mattyp.eq.FLUID) then
            call cset(cs,0,nd,NSD_ALL,INTERNAL)
            do nset=1,cs%no
                do IC=cs%ns(nset),cs%ne(nset)
                    c_old2(IC) = c(IC,iXact)
                enddo
            enddo
        endif
    ENDDO

    itNum = 0
25   continue
    itNum = itNum + 1
C   Loop through all of the cells
    DO nd=1,doma_no
        if (doma(nd)%mattyp.eq.FLUID) then
            call cset(cs,0,nd,NSD_ALL,INTERNAL)
            do nset=1,cs%no
                do IC=cs%ns(nset),cs%ne(nset)
C   Replace current value with correct value as recorded in c_old2
                    c(IC,iXact) = c_old2(IC)
                    c_old(IC) = c(IC,iXact)
                enddo
            enddo
        endif
    ENDDO

    do nd=1,doma_no
        if (doma(nd)%mattyp.eq.FLUID) then
            call cset(cs,0,nd,NSD_ALL,HALO)
            do nset=1,cs%no
                do IC=cs%ns(nset),cs%ne(nset)
                    c_old(IC) = c(IC,iXact)
                enddo
            enddo
        endif
    enddo

c   If this is a parallel run, swap/assign repeatedly until the values
c   stop changing
    if (parrun) then
        needSwap = .FALSE.
        do nd=1,doma_no
            call arrSwap(c,nd,2,iXact,FLUID)
        end do
        do nd=1,doma_no
            call cset(cs,0,nd,NSD_ALL,INTERNAL)

```

```

        do nset=1,cs%no
            do IC=cs%ns(nset),cs%ne(nset)
c      If the field values have changed, repeat the swap
            if (c(IC,iXact).ne.c_old(IC).AND.itNum.lt.itMax) then
                needSwap = .TRUE.
                goto 27
            endif
        end do
    end do
end do
end if
27  continue
    call LGLOR(needSwap,needSwapG)
    if(needSwapG) goto 25
C   Clear out c_old array
    DO nd=1,doma_no
        if(doma(nd)%mattyp.eq.FLUID) then
            call cset(cs,0,nd,NSD_ALL,INT_HALO)
            do nset=1,cs%no
                do IC=cs%ns(nset),cs%ne(nset)
                    c_old(IC) = 0
                    c_old2(IC) = 0
                enddo
            enddo
        endif
    ENDDO

C   Organize New Thrombi
        itNum = 0
    30  continue
        itNum = itNum + 1
        DO nd=1,doma_no
            if(doma(nd)%mattyp.eq.FLUID) then
                call cset(cs,0,nd,NSD_ALL,INTERNAL)
                do nset=1,cs%no
                    do IC=cs%ns(nset),cs%ne(nset)
                        IF((c(IC,inBldSt).EQ.2.OR.c(IC,inBldSt).EQ.3).AND.c(IC
$                          ,iThrNum).LT.1.0) THEN
C      Assign Thrombus Number
C---First check for thrombus neighbors
                            do nf=1,c2f(IC)%No
                                IS = c2f(IC)%List(nf)
C      Ignore negative (boundary) faces
                                IF(IS.GT.0) THEN
                                    do nfs=1,2
                                        IF(lfc(nfs,IS).NE.IC) THEN
                                            NBRCLL = LFC(nfs,IS)
C      --- If the neighbor cell is a thrombus, adopt its ThrNum
                                        IF(c(NBRCLL,iThrNum).GT.0.AND.ICLMAP(NBRCLL).GT.0.AND.
$                                          c(IC,iThrNum).EQ.0.AND.ICLMAP(IC).GT.0) THEN
                                            c(IC,iThrNum) = c(NBRCLL,iThrNum)
                                        ENDIF
                                    ENDIF
                                enddo
                            ENDIF
                        enddo
                    ENDIF
                enddo
            enddo
        ENDDO
    enddo

```

```

C   For a new thrombus without a thrombus neighbor, assign an ID by
C   choosing the prostar cell number of the star index
      IF(c(IC,iThrNum).EQ.0) THEN
        c(IC,iThrNum) = ICLMAP(IC)
        ThrLg(c(IC,iThrNum)) = .TRUE.
        do row=1,NCELL
          if(ThrLoc(row).EQ.0) then
            ThrLoc(row) = c(IC,iThrNum)
            goto 32
          endif
        enddo
32      continue
        ThrLg(c(IC,iThrNum)) = .TRUE.
      ENDIF

      ENDIF
C   populate c_old array for comparison
      c_old(IC) = c(IC,iThrNum)
    enddo
  enddo
endif
ENDDO
do nd=1,doma_no
  call cset(cs,0,nd,NSD_ALL,HALO)
  do nset=1,cs%no
    do IC=cs%ns(nset),cs%ne(nset)
      c_old(IC) = c(IC,iThrNum)
    enddo
  enddo
enddo

c   If this is a parallel run, swap/assign repeatedly until the values
c   stop changing
  if (parrun) then
    needSwap = .FALSE.
    do nd=1,doma_no
      call arrSwap(c,nd,2,iThrNum,FLUID)
    enddo
    do nd=1,doma_no
      call cset(cs,0,nd,NSD_ALL,INT_HALO)
      do nset=1,cs%no
        do IC=cs%ns(nset),cs%ne(nset)
c   If the field values have changed, report inconsistency, and repeat
c   the swap
          if (c(IC,iThrNum).ne.c_old(IC).AND.itNum.lt.itMax) then
            needSwap = .TRUE.
C   If a halo cell was given a thrombus number by the swap function,
C   then update the ThrLgH(nT) array with a .TRUE. value
          IF (c(IC,iThrNum).GT.0.AND.c_old(IC).EQ.0.AND.
$           ICLMAP(IC).LT.0) THEN
            ThrLgH(c(IC,iThrNum)) = .TRUE.
          ENDIF
          goto 35
        endif
      end do
    end do
  end do
end do

```

```

        end if
35      continue
        call LGLOR(needSwap,needSwapG)
        if(needSwapG) goto 30
C      Clear out c_old array
        DO nd=1,doma_no
          if(doma(nd)%mattyp.eq.FLUID) then
            call cset(cs,0,nd,NSD_ALL,INT_HALO)
            do nset=1,cs%no
              do IC=cs%ns(nset),cs%ne(nset)
                c_old(IC) = 0
              enddo
            enddo
          endif
        ENDDO

C      Communicate the thrombi that were created in this step and add them
C      to the thrombus array - ThrArray()
        numZero = 0
        DO row=1,NCELL
36      continue
          nT = IGMAX(ThrLoc(row))
          if(nT.EQ.0) then
            numZero = numZero+1
          else
            numZero = 0
            DO row1=1,GNCELL
              if(ThrArray(row1).EQ.0) then
                ThrArray(row1) = nT
                ThrMAP(nT) = row1
                IF(ThrArray(GNCELL).LT.row1) Then
                  ThrArray(GNCELL)=row1
                EndIf
                IF(ThrLoc(row).EQ.nT) ThrLoc(row) = 0
                goto 36
              endif
            ENDDO
          endif
C      Indicates 2 consecutive rows of zeroes in all ThrLoc arrays
        if(numZero.EQ.2) goto 38
        ENDDO
38      continue

        ENDIF

C      END EXECUTION AT THE BEGINNING OF ALL TIME STEPS
C#####

        ELSEIF(level.EQ.2) THEN

C#####
C      BEGIN EXECUTION AT END OF ALL TIME STEPS

C      Get the total volume of thrombus in the model
        vtot = 0
        DO nd=1,doma_no
          if(doma(nd)%mattyp.eq.FLUID) then

```

```

call cset(cs,0,nd,NSD_ALL,INTERNAL)
do nset=1,cs%no
  do IC=cs%ns(nset),cs%ne(nset)
    IF(c(IC,inBldSt).EQ.2.OR.c(IC,inBldSt).EQ.3) THEN
      vtot = vtot + VOL(IC)
    ENDIF
  enddo
enddo
endif
ENDDO
vtot = GSUM(vtot)
write(vThrFile,*) 'thr', TIME, vtot

```

C Fill in ThrLgH array

```

DO nd=1,doma_no
  if(doma(nd)%mattyp.eq.FLUID) then
    call cset(cs,0,nd,NSD_ALL,HALO)
    do nset=1,cs%no
      do IC=cs%ns(nset),cs%ne(nset)
        IF(c(IC,iThrNum).GT.0) THEN
          IF(ThrLgH(c(IC,iThrNum)).EQ..FALSE.) THEN
            ThrLgH(c(IC,iThrNum)) = .TRUE.
          ENDIF
        ENDIF
      enddo
    enddo
  endif
ENDDO

```

C LOOP THROUGH ALL THROMBI

```

do row=1,ThrArray(GNCELL)
  nT=ThrArray(row)
  IF(nT.EQ.0) goto 40
  CALL LGLOR(ThrMP(nT),ThrMPG)
  shearfc = 0
  height = 0
  nT1 = 0
  MAX = 0
  MAXNBR = 0
  MAXROW = 0
  ncount = 0
  SA1 = 0
  Sathromb = 0
  sheartot = 0
  surfarea = 0
  threshold = 0
  Thrneighbor = 0
  vthromb = 0
  CmbThr = .FALSE.
  EmbThr = .FALSE.
  do row1 = 1, ThrArray(GNCELL)
    ThrSA(row1) = 0.0
  enddo
  do row1 = 1, NeArray(NCELL)
    NeArray(row1) = 0
  enddo
  NeArray(NCELL) = 0

```

```

C   For each thrombus, determine number of cells contained, volume
C   of the thrombus, stress on the thrombus, and the number of cells
C   from each neighboring thrombus touching the reference thrombus.
      IF(ThrLg(nT)) THEN
        do nd=1,doma_no
          if(doma(nd)%mattyp.eq.FLUID) then
            call cset(cs,0,nd,NSD_ALL,INTERNAL)
            do nset=1,cs%no
              do IC=cs%ns(nset),cs%ne(nset)
                IF(c(IC,iThrNum).EQ.nT) THEN
                  ncount = ncount + 1
                  vthromb = vthromb + VOL(IC)
                  do nf=1,c2f(IC)%No
                    IS = c2f(IC)%List(nf)
                    IF(IS.GT.0) THEN
                      do nfs=1,2
                        IF(lfc(nfs,IS).NE.IC) THEN
                          NBRCLL = LFC(nfs,IS)
C   If the neighbor is not a thrombus, add to SA1
                          if(c(NBRCLL,iThrNum).EQ.0) THEN
                            SA1 = SA1 + sqrt(sv(1,IS)**2+sv(2,IS)**2+sv(3,IS)
$                               **2)
                            sheartot = sheartot + c(NBRCLL,iShrRt)*sqrt(sv(1,IS)
$                               **2+sv(2,IS)**2+sv(3,IS)**2)
                          EndIf
C   If the neighbor is a thrombus, add to SATHromb and record it in ThrSA
                          if(abs(c(NBRCLL,iThrNum)).GT.1E-4) THEN
                            if(abs(c(NBRCLL,iThrNum)-c(IC,iThrNum)).GT.1E-4) THEN
                              SATHromb = SATHromb + sqrt(sv(1,IS)**2+sv(2,IS)**2
$                               +sv(3,IS)**2)
                              nT1 = c(NBRCLL,iThrNum)
                              do row1=1,NCELL
                                if(ThrLoc(row1).EQ.0) then
                                  ThrLoc(row1) = nT1
                                  if(nT.EQ.myThromb) write(trkFile,*)
$                                   'ThrLoc.row = ',row1
                                  exit
                                elseif(ThrLoc(row1).EQ.nT1) then
                                  exit
                                endif
                              enddo
                              row1 = ThrMAP(nT1)
                              ThrSA(row1) = ThrSA(row1) + sqrt(sv(1,IS)**2+sv(2
$                               ,IS)**2+sv(3,IS)**2)
                            EndIf
                          EndIf
                        ENDIF
                      enddo
                    ENDIF
                  enddo
                ENDIF
              enddo
            ENDIF
          enddo
        EndIf
      EndIf
C   Calculate the surface area

```

```

do nr=0,regi_no
  if(regi(nr)%type.EQ.WALL) then
    call FsetBoun(fs,0,ND_ALL,NSD_ALL,nr,ISIDE_ALL)
    do nset=1,fs%no
      do IS=fs%ns(nset), fs%ne(nset)
        IC = lfc(1,IS)
        if(abs(c(IC,iThrNum)-nT).LT.1E-4) then
          surfarea = surfarea + sqrt(sv(1,IS)**2+sv(2,IS)**2+sv(3
$           ,IS)**2)
        endif
      enddo
    enddo
  endif
enddo
ENDIF

```

C Create a short list of thrombi that contact the thrombus nT

```

numZero = 0
DO row1=1,NCELL
39  continue
  nT1 = IGMAX(ThrLoc(row1))
  if(nT1.EQ.0) then
    numZero = numZero+1
  else
    numZero = 0
    DO row2=1,NCELL
      if(NeArray(row2).EQ.0) then
        NeArray(row2) = nT1
        IF(NeArray(NCELL).LT.row2) Then
          NeArray(NCELL)=row2
        EndIf
        IF(ThrLoc(row1).EQ.nT1) ThrLoc(row1) = 0
        goto 39
      elseif(NeArray(row2).EQ.nT1) then
        IF(ThrLoc(row1).EQ.nT1) ThrLoc(row1) = 0
        goto 39
      endif
    ENDDO
  endif

```

C Indicates 2 consecutive rows of zeroes in all ThrLoc arrays

```

if(numZero.EQ.2) exit
ENDDO

```

```

IF(ThrMPG) THEN

```

C Combine the values from all the processors

```

ncount = IGSUM(ncount)
vthromb = GSUM(vthromb)
SA1 = GSUM(SA1)
sheartot = GSUM(sheartot)
Sathromb = GSUM(Sathromb)
surfarea = GSUM(surfarea)
do row1=1,NeArray(NCELL)
  row2=ThrMAP(NeArray(row1))
  IF(nT.EQ.myThromb) then
    write(trkFile,*) IHPC,'row1 = ',row1
    write(trkFile,*) IHPC,'NeArray() = ',NeArray(row1)
    write(trkFile,*) IHPC,'row2 = ',row2
  endif
enddo

```

```

        write(trkFile,*) IHPC,'ThrSA() = ',ThrSA(row2)
        write(trkFile,*) IHPC,'GSUM = ',GSUM(ThrSA(row2))
    ENDIF
    SAsum = GSUM(ThrSA(row2))
    ThrSA(row2) = SAsum
    IF(nT.EQ.myThromb) write(trkFile,*) IHPC, 'ThrSA() = '
$      ,ThrSA(row2)
    enddo
ENDIF

C   Among the neighboring thrombi, which has more area touching
C   the reference thrombus?
    IF(ThrLg(nT).OR.ThrLgH(nT)) THEN
        IF(ThrLg(nT)) THEN
            do row1=1,NeArray(NCELL)
                row2 = ThrMAP(NeArray(row1))
                If(MAX.LT.ThrSA(row2)) Then
                    MAX = ThrSA(row2)
                    MAXNBR = ThrArray(row2)
                    MAXROW = row2
                Endif
            enddo
        ENDIF
    ENDIF

C   If >CoalArea of thrombus surface area touches the neighbor that
C   has the most touching surface area, join that neighbor
    IF(MAX.GT.0.0) THEN
        percCA = MAX/(Sathromb+SA1+surfarea)
        if(percCA.GT.CoalArea) then
            Thrneighbor = MAXNBR
            CmbThr = .TRUE.
        endif
    ENDIF
ENDIF
ENDIF

call LGLOR(CmbThr,CmbThrG)

IF(CmbThrG) THEN
    ThrArray(row) = 0
    ThrMAP(nT) = 0
    Thrneighbor = IGMAX(Thrneighbor)
    IF(ThrLg(nT).OR.ThrLgH(nT)) THEN
        IF(ThrLg(nT)) THEN
            ThrLg(Thrneighbor) = .TRUE.
        ENDIF
        IF(ThrLgH(nT)) THEN
            ThrLgH(Thrneighbor) = .TRUE.
        ENDIF
    ENDIF
ENDIF

C   If the accepting thrombus is in a Halo Cell, mark it as a
C   Multiprocessor Thrombus
    IF((ThrLg(nT).OR.ThrLg(Thrneighbor)).AND.(ThrLgH(nT).OR.
$      ThrLgH(Thrneighbor)).AND..not.ThrMP(Thrneighbor)) THEN
        ThrMP(Thrneighbor) = .TRUE.
    ENDIF
    do nd=1,doma_no
        if(doma(nd)%mattyp.eq.FLUID) then
            IF(ThrLg(nT).AND.ThrLgH(nT)) THEN

```



```

        call cset(cs,0,nd,NSD_ALL,INT_HALO)
    ELSEIF(ThrLg(nT)) THEN
        call cset(cs,0,nd,NSD_ALL,INTERNAL)
    ELSEIF(ThrLgH(nT)) THEN
        call cset(cs,0,nd,NSD_ALL,HALO)
    ENDIF
    do nset=1,cs%no
        do IC=cs%ns(nset),cs%ne(nset)
C   If the cell is in the combining thrombus, change the iThrNum
            IF(c(IC,iThrNum).EQ.nT) THEN
                c(IC,iThrNum) = Thrneighbor
            ENDIF
        enddo
    enddo
    endif
    enddo
    ENDIF
    ThrLg(nT) = .FALSE.
C   If the thrombus was combined with another thrombus, then we don't
C   need to run through the embolization routine for this thrombus
C   number.
        goto 40
    ENDIF

C   if the thrombus exists in this processor as an internal or halo cell
    IF(ThrLg(nT).OR.ThrLgH(nT)) THEN
C   Determine threshold for embolization
C   -- The shearfc here is the shear force (N) over the whole thrombus
        shearfc = sheartot*BldVisc
C   Assuming a hemisphere . . .
        height = 1.5 * vthromb/surfarea
        IF(height.LT.0.0001) THEN
            Threshold = AdhStrs*surfarea
        ELSE IF(height.GT.0.0001) THEN
C   Apply the criterion of Basmadjian
            Threshold = AdhStrs*surfarea*(height*10000)**(-0.875)
        ENDIF
C   Embolization Routine
        IF(shearfc.GT.Threshold) THEN
            EmbThr = .TRUE.
        ENDIF
    ENDIF

    call LGLOR(EmbThr,EmbThrG)

    IF(EmbThrG) THEN
        ThrArray(row) = 0
        ThrMAP(nT) = 0
        IF(ThrLg(nT).OR.ThrLgH(nT)) THEN
            IF(ThrLg(nT)) write(embFile,*) 'emb ',TIME,nT,ncount,vthromb
            do nd=1,doma_no
                if(doma(nd)%mattyp.eq.FLUID) then
                    IF(ThrLg(nT).AND.ThrLgH(nT)) THEN
                        call cset(cs,0,nd,NSD_ALL,INT_HALO)
                    ELSEIF(ThrLg(nT)) THEN
                        call cset(cs,0,nd,NSD_ALL,INTERNAL)
                    ELSEIF(ThrLgH(nT)) THEN

```

```

        call cset(cs,0,nd,NSD_ALL,HALO)
    ENDIF
    do nset=1,cs%no
        do IC=cs%ns(nset),cs%ne(nset)
            IF(c(IC,iThrNum).EQ.nT) THEN
C      If the cell belongs to the thrombus being embolized
                c(IC,iyPl) = 0.0
                c(IC,iyaPl) = 0.0
                c(IC,iThrFr) = 0.0
                c(IC,iThrNum) = 0.0
                c(IC,ixMact) = 0.0
                c(IC,ipulse) = 0.0
                c(IC,iactFr) = 0.0
                c(IC,ixM) = 0.0
                c(IC,inPlsFl) = 0.0
                c(IC,iSp) = 1.0
                c(IC,iPlSrc) = 0.0
                c(IC,iaPlSrc) = 0.0
                c(IC,iArThr) = 0.0

C      Calculate the new average thrombus volume
                c(IC,iEmbVol) = (c(IC,iEmbVol)*c(IC,iNumEmb)
$                   +vthromb) / (c(IC,iNumEmb) + 1)

C      Add 1 to the embolization count
                c(IC,iNumEmb) = c(IC,iNumEmb) + 1

C      Assume cells from which thrombus has embolized become
C      thrombus neighbors (corrected later)
                IF(c(IC,inBldSt).EQ.2.0) THEN
                    c(IC,inBldSt) = 5.0
                ELSE IF(c(IC,inBldSt).EQ.3.0) THEN
                    c(IC,inBldSt) = 4.0
                ENDIF
            ENDIF
        enddo
    endif
enddo
ENDIF

C      Flag that embolizing thrombus no longer exists in any processor
    ThrLg(nT) = .FALSE.
    ThrLgH(nT) = .FALSE.
    ThrMP(nT) = .FALSE.
ENDIF

40      continue
      enddo

C      Consolidate the ThrArray() entries
    Do row=1,ThrArray(GNCELL)
        if(ThrArray(row).EQ.0) then
            DO row1=ThrArray(GNCELL),row+1,-1
                if(ThrArray(row1).NE.0) then
                    ThrArray(row) = ThrArray(row1)
                    ThrMAP(ThrArray(row)) = row
                    ThrArray(row1) = 0
                endif
            enddo
        endif
    enddo

```

```

        goto 42
    endif
    ENDDO
    goto 43
endif
42  continue
    Enddo
43  continue

C  Update the recorded last filled row in the ThrArray
    intMax = ThrArray(GNCELL)
    Do row=ThrArray(GNCELL),1,-1
        if(ThrArray(row).EQ.0) then
            intMax = row-1
        else
            goto 45
        endif
    Enddo
45  continue
    ThrArray(GNCELL) = intMax

C  Check/correct thrombus-neighbor status
    itNum = 0
50  continue
    itNum = itNum + 1
    do nd=1,doma_no
        if(doma(nd)%mattyp.eq.FLUID) then
            call cset(cs,0,nd,NSD_ALL,INTERNAL)
            do nset=1,cs%no
                do IC=cs%ns(nset),cs%ne(nset)
                    if(c(IC,inBldSt).EQ.4.0.OR.c(IC,inBldSt).EQ.5.0) THEN
                        c(IC,iArThr) = 0.0
C  Loop through all faces of the cell
                            do nf=1,c2f(IC)%No
                                IS = c2f(IC)%List(nf)
C  Ignore negative (boundary) faces
                                    IF(IS.GT.0) THEN
                                        do nfs=1,2
                                            IF(lfc(nfs,IS).NE.IC) THEN
                                                NBRCLL = LFC(nfs,IS)
C  When a neighboring cell is actually a thrombus
                                                    if(c(NBRCLL,iThrNum).GT.0) then
                                                        c(IC,iArThr) = c(IC,iArThr) + sqrt(sv(1,IS)**2+sv(2
$
                                                            ,IS)**2+sv(3,IS)**2)
                                                    endif
                                                ENDIF
                                            enddo
                                        ENDIF
                                    enddo
                                enddo
                            if(c(IC,iArThr).EQ.0.0) then
                                IF(c(IC,inBldSt).EQ.4.0) THEN
                                    c(IC,inBldSt) = 0.0
                                    c(IC,iThrFr) = 0.0
                                ELSE IF(c(IC,inBldSt).EQ.5.0) THEN
                                    c(IC,inBldSt) = 1.0
                                ENDIF
                            endif
                        enddo
                    enddo
                enddo
            enddo
        endif
    enddo

```

```

        endif
    endif
    c_old(IC) = c(IC,iArThr)
    c_old2(IC) = c(IC,inBldSt)
enddo
enddo
endif
enddo
do nd=1,doma_no
    call cset(cs,0,nd,NSD_ALL,HALO)
    do nset=1,cs%no
        do IC=cs%ns(nset),cs%ne(nset)
            c_old(IC) = c(IC,iArThr)
            c_old2(IC) = c(IC,inBldSt)
        enddo
    enddo
enddo
enddo
c If this is a parallel run, swap/assign repeatedly until the values
c stop changing
    if (parrun) then
        needSwap = .FALSE.
        do nd=1,doma_no
            call arrSwap(c,nd,2,iArThr,FLUID)
            call arrSwap(c,nd,2,inBldSt,FLUID)
            call arrSwap(c,nd,2,iThrFr,FLUID)
        end do
        do nd=1,doma_no
            call cset(cs,0,nd,NSD_ALL,INT_HALO)
            do nset=1,cs%no
                do IC=cs%ns(nset),cs%ne(nset)
c If the field values have changed, repeat the swap
                    if (c(IC,iArThr).ne.c_old(IC).AND.itNum.lt.itMax) then
                        needSwap = .TRUE.
                        goto 55
                    endif
                    if (c(IC,inBldSt).ne.c_old2(IC).AND.itNum.lt.itMax) then
                        needSwap = .TRUE.
                        goto 55
                    endif
                end do
            end do
        end do
    end if
55    continue
    call LGLOR(needSwap,needSwapG)
    if(needSwapG) goto 50
C Clear out c_old arrays
    DO nd=1,doma_no
        if(doma(nd)%mattyp.eq.FLUID) then
            call cset(cs,0,nd,NSD_ALL,INT_HALO)
            do nset=1,cs%no
                do IC=cs%ns(nset),cs%ne(nset)
                    c_old(IC) = 0
                    c_old2(IC) = 0
                enddo
            enddo
        enddo
    enddo
endif

```

```
ENDDO
C END EXECUTION AT END OF ALL TIME STEPS
C #####
  ENDF
c
  end
```