THE HEMODYNAMICS DURING THROMBOSIS AND IMPACT ON THROMBUS GROWTH

A Dissertation
Presented to
The Academic Faculty

by

David Lawrence Bark Jr

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mechanical Engineering/Bioengineering

Georgia Institute of Technology
December 2010

THE HEMODYNAMICS DURING THROMBOSIS AND IMPACT ON THROMBUS GROWTH

Approved by:

Dr. David N. Ku, Advisor
The George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Dr. Cheng Zhu
The George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Dr. Larry McIntire
The Wallace H. Coulter Department of Biomedical Engineering
Georgia Institute of Technology

Dr. Luca Gerardo-Giorda
Department of Biology
Emory University

Dr. Kenichi Tanaka
School of Medicine
Emory University

Date Approved: November 11, 2010

DEDICATION

This thesis is dedicated to relatives who have passed away because of thrombosis and those who are currently at risk of developing thrombosis. I also dedicate this to the rest of my family and friends whom have always been supportive of my endeavors.

ACKNOWLEDGEMENTS

I wish to thank my advisor, Dr. David Ku, for his guidance and support during my doctoral work. I also wish to thank Dr. Ku's various lab members that have worked with me during my years at Georgia Tech. Additionally, I'd like to thank past and present members of Wing 2A, especially David Murphy, William Wan, and Choon Hwai Yap. Finally, I would like to thank family and friends for their love, support, and patience.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

NOMENCLATURE

Symobol	Description	
st	Ch 2 pg. 14	Stenosis severity by diameter
d_i	Ch 2 pg. 14	Minimum lumen diameter
d_0	Ch 2 pg. 14	Nominal lumen diameter
r	Ch 2 pg. 14	Radius of vessel
x	Ch 2 pg. 14	Axial coordinate
x_0	Ch 2 pg. 14	Stenosis half length
Δp	Ch 2 pg. 19	Pressure differential
μ	Ch 2 pg. 19	Kinematic viscosity
L	Ch 2 pg. 19	Straight vessel length upstream and downstream of stenosis
$\dot{\gamma}$	Ch 2 pg. 19	Shear rate
R	Ch 2 pg. 19	Distal vessel resistance
r_0	Ch 2 pg. 19	Nominal vessel radius
Q	Ch 2 pg. 20	Volume flow rate
τ	Ch 2 pg. 21	Shear stress

t	Ch 2 pg. 21	Time
v	Ch 2 pg. 21	Velocity
T	Ch 2 pg. 21	Time interval along a curve
y'	Ch 3 pg. 46	Actual stenosis radius
y	Ch 3 pg. 46	Observed stenosis radius
R_0	Ch 3 Pg. 46	Outer physical radius of vessel
n_{air}	Ch 3 pg. 46	Refractive index of air
n_{pyrex}	Ch 3 pg. 46	Refractive index of pyrex
$n_{thrombosis}$	Ch 3 pg. 47	Refractive index of thrombosis
R_i	Ch 3 pg. 47	Initial lumen radius of vessel
τ	Ch 3 pg. 51	Thrombus thickness
J	Ch 3 pg. 51	Thrombus groth rate
$t_{occlusion}$	Ch 3 pg. 53	Occlusion time
$T_{lag\ time}$	Ch 3 pg. 53	Lag time
C_j	Ch 4 pg. 80	Concentration of species j
J_j	Ch 4 pg. 80	Flux of species j

S_j	Ch 4 pg. 80	Source/sink of species j
$\vec{v}$	Ch 4 pg. 81	Velocity vector
D_j	Ch 4 pg. 81	Diffusivity of species j
ψ	Ch 4 pg. 81	Field potential of species j
ζ_j	Ch 4 pg. 82	Drift term
k_t	Ch 4 pg. 86	Platelet to surface kinetic binding rate
D_{th}	Ch 4 pg. 88	Thermal diffusivity
a	Ch 4 pg. 88	Particle diameter
ϕ	Ch 4 pg . 88	Hematocrit
k_B	Ch 4 pg. 88	Boltzmann constant
T	Ch 4 pg. 88	Temperature
R_{eq}	Ch 4 pg. 88	Equivalent radius of a sphere
Q_0	Ch 4 pg. 91	Nominal flow rate

SUMMARY

Atherothrombosis can induce acute myocardial infarction and stroke by progressive stenosis of a blood vessel lumen to full occlusion. The goal of the present work is to characterize the hemodynamics, specifically wall shear rates, that are pertinent to an occluding blood vessel. We also aim to characterize the rate of thrombus growth and how thrombus growth varies with wall shear rates. To further investigate the mechanisms behind thrombus growth, we develop a species transport model to evaluate rate-limiting factors of thrombus growth and to extend the model to predict thrombus growth.

Computational studies of severely stenotic idealized vessels were performed to investigate the wall shear rates that may exist. The study shows that maximum shear rates in severe short stenoses were found to exceed 250,000 s^{-1} (9,500 dynes/cm2). We utilize an *in vitro* experiment consisting of blood flow through a collagen coated stenosis to study the rate of thrombus growth. Computational fluid dynamics are used to determine shear rates along the thrombus surface as it grows. We found a strong positive correlation between thrombus growth rates and shear rates up to 6,000 s^{-1} after a log-log transformation ($r=0.85$, $p<0.0001$). Growth rates at pathologic shear rates were typically 2-4 times greater than for physiologic shear rates below 400 s^{-1}. To determine the rate-limiting factors of thrombus growth, a computational model of platelet transport and thrombus growth was developed. We show that thrombus is transport rate-limited for shear rates below 6,000 s^{-1}, while it is more likely to be kinetic rate-limited for higher

shear rates. Predictions of occlusion times based on the model demonstrate that increases in stenosis severity results in decreased time to occlusion.

CHAPTER 1. INTRODUCTION

This dissertation explores the hemodynamic characteristics associated with atherothrombosis, a term coined to characterize thrombosis (blood clot) that forms near an atherosclerotic lesion in the arterial system. Organization of this dissertation begins with a background that sets up a hypothesized process for thrombosis based on current literature. Chapter 2 characterizes the hemodynamics that may exist in an occluding coronary artery. The wall shear is found to be higher than previously considered, which sets up subsequent experiments that focus on high shear over severely stenotic arteries. Chapter 3 evaluates thrombus progression over a stenosis under high shear conditions. An analytical model of the process is explored in Chapter 4. Chapter 2-4 are laid out in a self-contained manner with a introduction, description of methods, results, and discussion. Chapter 5 ties together the new information discovered in this dissertation and relates it to the previous literature. Implications of this dissertation are then explored with a subsequent explanation of future work for this area of research.

Clinical Significance of Thrombosis

Heart disease remains the leading cause of natural death in the United States, with the majority of these deaths occurring from acute coronary events. Acute coronary events have been linked to acute occlusive atherothrombosis that creates a progressive stenosis and can induce sudden stroke, myocardial infarction, and limb necrosis by restricting blood flow. Atherothrombosis is found on or near an atheroma in 90% of the cases of acute myocardial infarction (AMI) through postmortem studies [1-4]. Rupture

of atherosclerotic plaque caps is implicated as a cause for atherothrombogenesis by exposing an adhesive surface for thrombus growth [5-6].

Atherothrombus is different from red thrombus that forms due to the coagulation cascade in low and stagnating flow regions. Instead, atherothrombosis occurs in regions of high flow and high shear rates. However, the hemodynamic conditions (shear rates) for severe stenoses are still unknown and thrombosis has historically been studied at shear rates on the order of 1,000 s^{-1} or below. Thrombosis is characteristically formed of interconnected networks of platelets and binding proteins [7]. Anticoagulants are not completely effective in treating patients at risk of atherothrombosis since its formation involves growth mechanisms that are different from coagulation. Antiplatelet therapies have partial effectiveness, but can lead to significant bleeding problems. Therefore, there is still a need for an effective therapy for treating patients that are at risk for atherothrombosis. A greater understanding of the thrombus process, including the rate-limiting mechanisms of thrombus growth, may elucidate new therapies.

Currently, patients receive prophylactic intervention when they develop substantial plaque in their arteries, causing a high degree stenosis. However, mechanisms that implicate risk in such plaques have yet to be completely elucidated. In addition thrombus has also been found on mild to moderate stenoses, indicating that stenosis severity is not sufficient for assessing thrombus risk over an atheroma. Furthermore, it is difficult to manage intra-arterial thrombosis once it initiates, because thrombus can completely occlude a vessel in less than 20 minutes [6, 7], stressing the importance of developing better predictive abilities.

Thrombosis and its Implications

Atherosclerotic plaques generally develop in specific arteries, forming a stenosis, or vessel constriction, that causes variation in local hemodynamics. A thrombotic event can be triggered in the region of an atheroma, progressing a stenosis as a consequence of thrombus growth. In the present study we consider the hemodynamics over a stenosis based on coronary artery conditions. Flow over a stenosis is controlled by pressure drop across the stenosis, distal arterioles, and the distal capillary bed. Reactive hyperemia occurs in the distal arterioles when a stenosis is present in order to maintain nominal flow rates through the coronary arteries. Reactive hyperemia can become saturated for severely stenotic lesions, resulting in a drop in flow rates until a flow is ceased, upon vessel occlusion. However, hemodynamics for severely stenotic arteries (>75% by diameter) are not known, even though the hemodynamics are thought to be important to growth of an atheroma and growth of thrombosis. Prevention of thrombosis on these very severe lesions can allow some flow to be maintained through the artery, mitigating the risk of ischemia.

Atherothrombus, or thrombus formation over an arterial atherosclerotic lesion, is mostly composed of platelets and binding proteins [7-9]. Platelets normally transport through blood vessels as inert particles at physiological shear rates when exposed only to vessel endothelium. Endothelial disruption typically results in hemostasis resulting from platelet deposition. However, thrombus can also form in regions of plaque [1-4], which is believed to occur where the atherosclerotic plaque ruptures. The ruptured region exposes the blood flow to the underlying subendothelial and medial layers of an

atheroma, instead of the usual endothelial layer. Thrombus can superimpose the plaque resulting in vessel occlusion causing a beneficial hemostasis function to turn into a pathological life threatening issue.

Initial platelet adherence can occur through collagen that is prevalent in the subendothelial layer, which allows direct platelet binding [10-11]. The process of platelet adherence to a surface is similar to the process of platelet-platelet aggregation, with many of the same major bridging glycoproteins [12]. Platelet affixation to glycoproteins is found to be shear dependent [13-14]. Glycoproteins can even undergo conformational changes, depending on the shear [15]. Another important phenomenon regarding platelet binding occurs upon platelet activation. Activation triggers a response from an additional platelet surface glycoprotein that is believed to increase the stability of platelet bonds to a surface. Activation is induced by synergistic activity of agonists, shear stress, and binding of glycoproteins.

Binding on the nano to micro-scale translates to platelet accumulation, which occurs most prevalently in the throat of the stenosis [9, 16-19]. Shear rates are the highest this region, reaching at least 80,000 s^{-1} in canines [20-22], which is more than 2 orders of magnitude higher than wall shear rates of the surrounding vessel (400 s^{-1}). However, thrombus accumulation has only been studied up to shear rates near 32,000 s^{-1} [23-25]. Studies that have been performed have focused on initial platelet adherence to a surface and are usually stopped after 5 minutes of blood perfusion. Therefore, there is little known about thrombus growth over previously deposited thrombus. Growth rates that have been found have an increasing trend relative to initial wall shear rates [13, 19, 24-31]. However, the interaction between shear and thrombus growth remains unknown

once thrombus begins protruding into the perfusion chambers. A relationship between a hemodynamic variable, such as shear rate, and thrombus growth rates provides information on favorable flow conditions for thrombus growth. The same hemodynamic conditions can be used to guide studies of transport and binding mechanisms that contribute to thrombus growth.

The binding and accumulation of platelets can not occur without the transport of platelets to the binding surface. Platelet binding can either be kinetic rate-limited if transport rates exceed kinetic binding rates, or platelets can be transport rate-limited. Reductions in the rate-limiting mechanisms would reduce thrombus growth rates. Transport of platelets is thought to be enhanced relative to thermal diffusivity predicted from the Stokes-Einstein equation. One method of enhancement is the increased convection thought to result from rotary motion of red blood cells [32-33] and increases in the collision rate as function proportional to shear [34]. In addition platelet concentration profiles become askew towards a blood vessel wall in flowing whole blood, known as margination [35-37]. However, platelet transport in whole blood is complex because platelets transport in a concentrated suspension of deformable particles. Multiple interactions occur at one time and the motion of an individual particle should exhibit statistical behavior similar to random walk processes, which is analogous to diffusion based on Brownian motion.

Research Goals and Hypotheses

Platelets are required to attach at shear rates found on severe stenoses for atherothrombosis to reach full occlusion, yet many thrombus studies have not considered shear rates greater than 30,000 s^{-1}, with most studies focused at shear rates of 1,000 s^{-1} or

lower. We quantify wall shear rates for severe stenoses to determine what hemodynamic conditions may exist as thrombus approaches occlusion of a vessel. We aim to answer the question: What is the maximum shear rate that can occur over a stenotic lesion based on coronary boundary conditions?

Wall shear rates have been linked to thrombus growth. We evaluate the interconnection between wall shear rates and thrombus growth rates by determining the piecewise correlation coefficients for shear rates ranging from 100 to 100,000 s^{-1}. The spatial and temporal distribution of thrombus is studied in relation to wall shear rates that are updated as thrombus grows. With this study we aim to answer the questions: What is the relationship between thrombus growth rates and wall shear rates, if there is one?

Thrombus grows at high rates. We investigate whether thrombus growth is kinetic or transport rate-limited for shear rates of 100-100,000 s^{-1} and whether or not there is a transition from one rate-limiting mechanism to another. Knowledge of the rate-limiting mechanism allows binding or transport to be targeted when designing medical devices and therapies for thrombosis prevention. A model of thrombus growth is also developed to predict occlusion times for various stenoses. A predictive model can be used to asses the thrombotic risk for a given stenosis. We aim to answer the question: Is there a range of shear where thrombus growth rates are transport rate-limited and is there a range where growth rates are rate-limited by the kinetic binding of platelets to thrombus?

Specific aims 1, 2, and 3 are used to develop answers to these questions and are presented in chapters 2, 3, and 4 respectively in a journal article format.

Specific Aims

1. Quantify maximum shear rates in pathologic arterial disease by CFD calculations over a stenosis with smoothly varying geometry for flow rates governed by a combination of stenosis flow resistance and distal flow resistance. "Wall Shear over High Degree Stenoses Pertinent to Atherothrombosis"

Since thrombus formation and growth may be shear-dependent, we quantify the magnitude of shear rates in idealized severely stenotic coronary arteries (≥ 75% by diameter) using computational fluid dynamics to characterize the shear environment that may exist during atherothrombosis. Stenosis length, height, and roughness are studied to determine what conditions result in the highest shear rates. Furthermore we investigate whether stenosis eccentricity or flow pulsatility has a strong effect on shear rates. Lastly, we compute "shear histories" of circulating platelets in these stenoses to determine if platelet activation is likely to occur.

2. Quantify the relationship between experimental thrombus growth rates and thrombus surface shear rates relative to space and time. "Effect of High Wall Shear on *In Vitro* Thrombus Growth over a Stenosis"

Local hemodynamics may strongly influence the development of thrombosis in a high shear situation. The current study characterizes the formation and progression of thrombosis over an existing stenosis in an in vitro system under pathophysiologic shear conditions. Thrombus growth is measured from optical backlighting of blood and edge detection techniques as thrombus progresses. Since wall shear rates change during thrombus growth, we compute the changing flow field during thrombosis by using computational fluid dynamics. Linear and nonlinear correlation coefficients are

investigated between thrombus growth rates and shear rates ranging from 100 to 100,000 s^{-1}.

3. Develop a convective transport model of platelet transport to the wall from increasing shear rates to compare arrival rates to thrombus growth rates in a stenosis. The transport model of platelets will include Enhanced Diffusivity, margination, and binding kinetic rate constants. "Rate-Limiting Thrombosis: A Study of Platelet Transport and Binding at Pathophysiological Shear

Platelet transport may play a substantial role in the rate of thrombus growth for a specific range of shear. A computational model is developed in the present study to investigate the role of transport in thrombus growth rates based on local hemodynamics and to extend the model to predict the time to occlusion for diseased arteries. The convection-diffusion equation is used to determine platelet transport based on an additional red blood cell term in the field potential. Once platelets transport to the thrombus surface, the thrombus is specified to grow based on a kinetic binding rate. Hemodynamics in the computational domain are updated as thrombus grows. Results are compared with experiments of thrombus growth under the same hemodynamic conditions to verify the model accuracy. The model is extended to predict thrombus occlusion time a 40, 60, and 80% stenosis.

References

1. Yutani, C., et al., *Coronary atherosclerosis and interventions: pathological sequences and restenosis.* Pathology International, 1999. **49**(4): p. 273-290.
2. Davies, M.J. and A. Thomas, *Thrombosis and acute coronary-artery lesions in sudden cardiac ischemic death.* N Engl J Med, 1984. **310**(18): p. 1137-40.
3. Viles-Gonzalez, J.F., V. Fuster, and J.J. Badimon, *Atherothrombosis: a widespread disease with unpredictable and life-threatening consequences.* European heart journal., 2004. **25**(14): p. 1197-1207.
4. Chandler, A., et al., *Coronary thrombosis in myocardial infarction:: Report of a workshop on the role of coronary thrombosis in the pathogenesis of acute myocardial infarction.* The American Journal Of Cardiology, 1974. **34**(7): p. 823-833.
5. Friedman, M. and G.J. Van den Bovenkamp, *The pathogenesis of a coronary thrombus.* The American Journal Of Pathology, 1966. **48**(1): p. 19-44.
6. Horie, T., M. Sekiguchi, and K. Hirosawa, *Relationship between myocardial infarction and preinfarction angina: a histopathological study of coronary arteries in two sudden death cases employing serial section.* American Heart Journal, 1978. **95**(1): p. 81-88.
7. Davies, M.J. and A.C. Thomas, *Plaque fissuring--the cause of acute myocardial infarction, sudden ischaemic death, and crescendo angina.* British Heart Journal, 1985. **53**(4): p. 363-373.
8. Wootton, D.M., *Mechanistic Modeling of Occlusive Arterial Thrombosis*, in *Mechanical Engineering*. 1998, Georgia Institute of Technology: Atlanta. p. 421.
9. Flannery, C.J., *Thrombus formation under high shear in arterial stenotic flow.*, in *Mechanical Engineering*. 2005, Georgia Tech: Atlanta.
10. Kunicki, T.J., et al., *Variability of integrin alpha 2 beta 1 activity on human platelets.* Blood., 1993. **82**(9): p. 2693-2703.
11. Saelman, E.U., et al., *Platelet adhesion to collagen types I through VIII under conditions of stasis and flow is mediated by GPIa/IIa (alpha 2 beta 1-integrin).* Blood., 1994. **83**(5): p. 1244-1250.
12. Vanhoorelbeke, K., et al., *Inhibition of platelet adhesion to collagen as a new target for antithrombotic drugs.* Current Drug Targets. Cardiovascular & Haematological Disorders, 2003. **3**(2): p. 125-140.
13. Savage, B., E. Saldívar, and Z.M. Ruggeri, *Initiation of platelet adhesion by arrest onto fibrinogen or translocation on von Willebrand factor.* Cell., 1996. **84**(2): p. 289-297.
14. Yago, T., et al., *Platelet glycoprotein Ibalpha forms catch bonds with human WT vWF but not with type 2B von Willebrand disease vWF.* The Journal Of Clinical Investigation, 2008. **118**(9): p. 3195-3207.
15. Schneider, S.W., et al., *Shear-induced unfolding triggers adhesion of von Willebrand factor fibers.* Proceedings Of The National Academy Of Sciences Of The United States Of America, 2007. **104**(19): p. 7899-7903.

16. Folts, J.D., E.B. Crowell, Jr., and G.G. Rowe, *Platelet aggregation in partially obstructed vessels and its elimination with aspirin.* Circulation., 1976. **54**(3): p. 365-370.
17. Badimon, L. and J.J. Badimon, *Mechanisms of arterial thrombosis in nonparallel streamlines: platelet thrombi grow on the apex of stenotic severely injured vessel wall. Experimental study in the pig model.* The Journal Of Clinical Investigation, 1989. **84**(4): p. 1134-1144.
18. Lassila, R., et al., *Dynamic monitoring of platelet deposition on severely damaged vessel wall in flowing blood. Effects of different stenoses on thrombus growth.* Arteriosclerosis : an official journal of the American Heart Association, Inc., 1990. **10**(2): p. 306-315.
19. Markou, C.P., et al. *Role of high wall shear rate on thrombus formation in stenoses*. in *Advances in Bioengineering*. 1993. New Orleans: ASME.
20. Bark, D. and D. Ku, *Wall shear over high degree stenoses pertinent to atherothrombosis.* Journal of Biomechanics, 2010.
21. Siegel, J.M., et al., *A scaling law for wall shear rate through an arterial stenosis.* Journal of Biomechanical Engineering, 1994. **116**(4): p. 446-451.
22. Strony, J., et al., *Analysis of shear stress and hemodynamic factors in a model of coronary artery stenosis and thrombosis.* American journal of physiology. Heart and circulatory physiology, 1993. **265**(5): p. H1787-1796.
23. Ku, D.N. and C.J. Flannery, *Development of a flow-through system to create occluding thrombus.* Biorheology, 2007. **44**(4): p. 273-284.
24. Barstad, R.M., P. Kierulf, and K.S. Sakariassen, *Collagen induced thrombus formation at the apex of eccentric stenoses--a time course study with non-anticoagulated human blood.* Thrombosis And Haemostasis, 1996. **75**(4): p. 685-692.
25. Barstad, R.M., et al., *A perfusion chamber developed to investigate thrombus formation and shear profiles in flowing native human blood at the apex of well-defined stenoses.* Arteriosclerosis And Thrombosis: A Journal Of Vascular Biology / American Heart Association, 1994. **14**(12): p. 1984-1991.
26. Markou, C.P., Lindahl, A.K., Siegel, J.M., Ku, D.N., Hanson, S.R., *Effects of blocking the platelet GPIb interaction with von Willebrand Factor under a range of shearing forces.* Annals Of Biomedical Engineering, 1993. **21**: p. 220.
27. Badimon, L., et al., *Influence of arterial damage and wall shear rate on platelet deposition. Ex vivo study in a swine model.* Arteriosclerosis (Dallas, Tex.), 1986. **6**(3): p. 312-320.
28. Hubbell, J.A. and L.V. McIntire, *Platelet active concentration profiles near growing thrombi. A mathematical consideration.* Biophysical journal., 1986. **50**(5): p. 937-945.
29. Alevriadou, B.R., et al., *Real-time analysis of shear-dependent thrombus formation and its blockade by inhibitors of von Willebrand factor binding to platelets.* Blood., 1993. **81**(5): p. 1263-1276.
30. Sakariassen, K.S., et al., *Collagen type III induced ex vivo thrombogenesis in humans. Role of platelets and leukocytes in deposition of fibrin.* Arteriosclerosis, Thrombosis, And Vascular Biology, 1990. **10**(2): p. 276-284.

31. Turitto, V.T. and H.R. Baumgartner, *Platelet deposition on subendothelium exposed to flowing blood: mathematical analysis of physical parameters.* Transactions - American Society for Artificial Internal Organs., 1975. **21**: p. 593-601.
32. Keller, K.H., *Effect of fluid shear on mass transport in flowing blood.* Federation proceedings., 1971. **30**(5): p. 1591-1599.
33. Goldsmith, H.L., *The flow of model particles and blood cells and its relation to thrombogenesis.* Progress In Hemostasis And Thrombosis, 1972. **1**: p. 97-127.
34. Goldsmith, H. and S. Mason, *The flow of suspensions through tubes. III. Collisions of small uniform spheres.* Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 1964. **282**(1391): p. 569-591.
35. Tilles, A.W. and E.C. Eckstein, *The near-wall excess of platelet-sized particles in blood flow: its dependence on hematocrit and wall shear rate.* Microvascular research., 1987. **33**(2): p. 211-223.
36. Aarts, P.A., et al., *Blood platelets are concentrated near the wall and red blood cells, in the center in flowing blood.* Arteriosclerosis : an official journal of the American Heart Association, Inc., 1988. **8**(6): p. 819-824.
37. Tangelder, G.J., et al., *Localization within a thin optical section of fluorescent blood platelets flowing in a microvessel.* Microvascular research., 1982. **23**(2): p. 214-230.

CHAPTER 2. SPECIFIC AIM 1: WALL SHEAR OVER HIGH DEGREE STENOSIS PERTINENT TO ATHEROTHROMBOSIS

Introduction

Heart disease remains the leading cause of natural death in the United States, with the majority of these deaths occurring from acute coronary events. Occluding thrombus, which restricts blood flow, is present over an atheroma in many of these cases [1]. Atherosclerotic plaques and thrombus form a stenosis, thus modifying the local hemodynamics including shear, which is known to affect thrombosis characteristics [2]. Platelet deposition rates have been found to increase with shear rates reaching 10,000 s^{-1} [3-6], with little data for higher shear. Shear correlated processes involved in atherothrombosis include mass transport [7], platelet activation [8], binding protein effectiveness [9-10], and dissociation rates [11]. The goal of the present study is to quantify the largest wall shear rates that may occur in a diseased coronary artery during the cardiac cycle to define the hemodynamic conditions for occlusive thrombosis.

Shear rates over a coronary artery atheroma are generally governed by the atheroma shape, ascending aorta pressure, coronary vein pressure, and coronary arteriole resistance. The maximum pressure drop between the aorta and the coronary veins is approximately 100 mmHg [12]. Coronary arterioles can vasodilate, reducing the vessel resistance by reactive hyperemia, which can saturate as a growing stenosis nears occlusion, such that nominal flow rates can no longer be maintained [13]. Full vasodilation is implemented during the clinical measurement of fractional flow reserve (FFR) to estimate stenosis severity, defined as the ratio of maximal myocardial flow in the presence of the stenosis relative to the absence of the stenosis. FFR is commonly

estimated by the ratio of pressure distal to a stenosis relative to the pressure proximal (aortic pressure), which monotonically decreases for severe stenoses that are greater than 50% [14-16].

Maximum shear rates in moderate stenoses scale with the square root of the Reynolds number [17]. The scaling was limited to stenoses up to 68% diameter, corresponding to a maximum shear rate of 72,000 s^{-1}. Shear rates are reported to reach as high as 84,000 s^{-1} for a 65% stenosis by diameter based on numerical studies of a canine model [18]. Comparatively, the highest shear rates typically found in the normal circulation occurs in the arterioles and can reach 4,700 s^{-1} [19].

Plaque geometrical morphology can vary substantially, which can affect shear and energy loss with plaque eccentricity for low to moderate stenoses [20-21]. However, [22] report minimal eccentricity effects on energy loss and shear for moderate to severe stenoses. They found length and stenosis severity to impact energy loss for the same stenoses. Unsteady effects have been found to have negligible impact on the location of peak shear rates over a stenosis apex [23].

Previous hemodynamic studies have not considered stenoses greater than 70%, as the focus has been on moderate stenoses. Prophylactic treatment is usually provided for higher stenosis severity due to surgical benefits found for >70% stenoses. However, moderate stenoses can be swiftly augmented by thrombosis until full vessel occlusion, creating severe stenoses of great clinical significance that can induce acute myocardial infarction. We quantify the hemodynamics for severe stenoses ($\geq$ 75%) through computational fluid dynamics. We consider geometrical variables of stenosis severity,

length, eccentricity, and roughness to quantify maximum wall shear, which provides information on near wall platelet biomechanics.

Methods

Stenosis severity in this study is represented as percent stenosis by diameter [24], *st*:

Equation 2.1

$$st = 100\frac{d_0 - d_i}{d_0}\%$$

where d_i is the minimum lumen diameter typically visible on angiography. A blood vessel was modeled as a rigid tube with a nominal diameter, d_0, of 3 mm to simulate a coronary artery. A 2-D, axisymmetric stenosis was created to represent an atheroma in a coronary sized vessel based on the function:

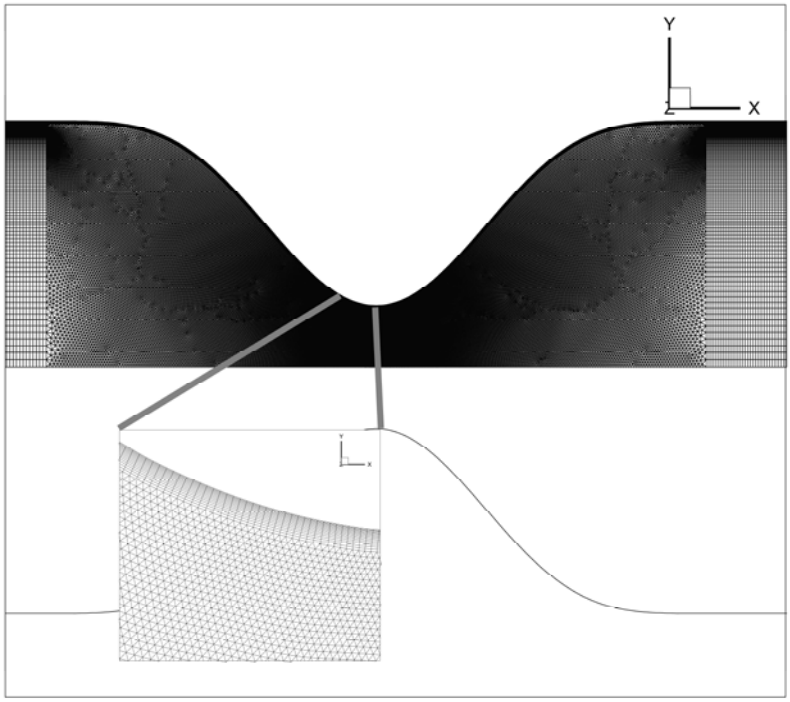
Equation 2.2

$$\frac{r}{d_0} = \frac{1}{2} - \frac{st}{800}\left[1 - \cos\left(\pi\left(\frac{x}{x_0} + 1\right)\right)\right]^2$$

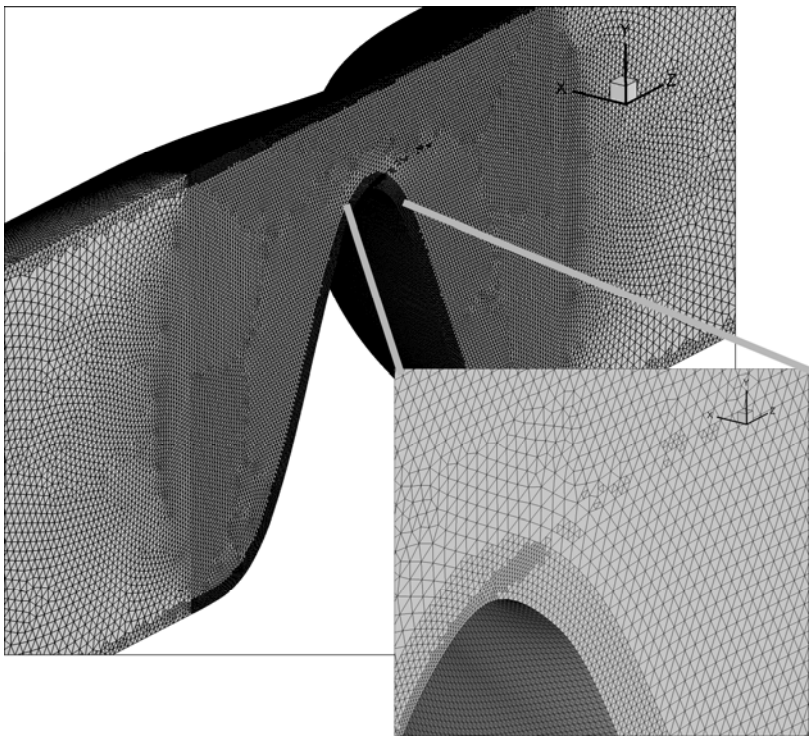
where x_0 is half of the stenosis length, and r is the radius at a given axial coordinate, x. The geometry generates a larger spatial gradient relative to the commonly used cosine curve [21]. A variation was used because *in vivo* stenoses are commonly very abrupt and streamlined, which may result in higher shear rates relative to the cosine geometry. A proximal length of 20 diameters, relative to the stenosis was included for flow development to yield a stable parabolic velocity profile. 230 is the maximum proximal Reynolds number for our study and therefore the theoretical flow development entrance length of 13.8 diameters would be the maximum required length for fully developed flow

to enter the region of the stenosis. A distal length of 30 diameters was included to allow flow to reattach, if separated on the stenosis, and yield a stable solution for the stenosis region of interest.

3-D symmetric eccentric stenoses were developed by generating a surface over a series of half circles. The centerline was defined as r_0-r, while the radius was defined by Equation 2.2, thus creating a lumen diameter equivalent to the axisymmetric stenosis at all axial locations. An example of the 2-D and 3-D mesh is shown in Figure 2.1A and B respectively, which were generated through Gambit 2.4.6 (ANSYS Santa Clara, CA).



A



B

Figure 2.1: 75% stenosis by diameter mesh for A) Case 2 (4 mm – 2-D axisymmetric) and B) Case 5 (4 mm –3-D eccentric).

A structured mesh was applied near the wall boundary to capture the boundary layer. Triangular elements were used in the stenosis region of the axisymmetric model, spanning an axial distance of $2x_0$. A structured mesh was applied proximal and distal to the stenosis, with cell density skewed toward the vessel wall. The mesh size varied slightly for each stenosis due to geometrical variation, but averaged near 400,000 computational cells. The Delaunay Triangulation method through TGrid was used to create triangles on surfaces and tetrahedrals to generate the mesh for the 3-D eccentric stenoses, averaging near 8 million tetrahedral cells. Mesh resolution was set higher near the wall relative to the central axis and in other regions of high shear in order to capture large velocity gradients, e.g., the fluid boundary layer. Mesh size independence was verified.

A rough, thrombus surface was created for one case by augmenting the stenosis, Equation 2.2, with a cubic spline to create an elongated stenotic profile that had the general shape of thrombus formed in *in vitro* experiments [5] and *ex vivo* experiments [25]. The spline was further modified by adding brown noise to the augmented surface, using an amplitude of 200 microns, to generate low frequency surface roughness.

Flow in this study is laminar and Newtonian. The Reynolds numbers remain sufficiently below the critical turbulent Reynolds number for the studied stenoses, based on extrapolation and interpolation [21]. The flow field was determined through the incompressible Navier-Stokes equations using the finite-volume based commercial computational fluid dynamics package, Fluent 6.3.26 (ANSYS Santa Clara, CA). Second order accurate schemes were used with SIMPLE velocity-pressure coupling. Solution

convergence was verified with a mass balance between the inlet and outlet. All residuals were below 10^{-4}.

The density of fluid was set at 1025 kg/m^3, with a constant blood viscosity of 3.8 cP. Fahraeus-Lindquist effects in small vessels would only be possible for the case of the 98% and 99% stenosis, where viscosity may reduce by approximately 25% for the 99% stenosis [26]. These effects were neglected for this study.

The boundary conditions used in this study are presented in Table 2.1 and are designed to estimate maximum shear rates reached in a coronary artery. Case 1 consisted of a pulsatile left coronary flow waveform [27] inlet. Only steady flow was considered for subsequent simulations, as the pulsatile flow study demonstrated the fluid behavior is quasi-steady. Case 2, the base control condition, simulates reactive hyperemia with a pressure differential of 100 mmHg and variable distal peripheral resistance, *R*, of 11 mmHg*s/ml, placed 5 mm proximal to the outlet. The non-stenotic Reynolds number is 230. Case 3 consists of a constant 50 mmHg pressure drop across the stenosis as a lower bound. Case 4 involves an elongated smooth stenosis; while, Case 5 illustrates a flow through a 3-D, eccentric stenosis. Case 6 simulates flow through a thrombus with a rough surface shape.

Table 2.1: Stenosis conditions

Case	Inlet Boundary	Stenosis Length	Stenosis Degree	Distal Resistance
1	Pulsatile Flow	4 mm	95%	n/a
2	100 mmHg	4 mm	50, 75, 85, 90, 93, 95, 97, 98, 99%	Y
3	50 mmHg	4 mm	75, 90, 95, 98, 99%	N
4	100 mmHg	16 mm	50, 75, 85, 90, 93, 95, 97, 98, 99%	Y
5	100 mmHg	4 mm	Eccentric 75, 90, 95, 98, 99%	Y
6	100 mmHg	4 mm	90% w/ Rough Surface w/ 97% max	Y

For comparison, maximum shear rates at the stenosis apex may be estimated from a 1-D viscous flow assumptions by applying a series of resistances, a variation on a previous model [28]. A summation of pressure loss terms include viscous loss due to the stenosis section, viscous loss due to straight vessel section spanning length, L, and the previously described peripheral resistance. If Hagen-Poiseuille flow is assumed, then our theoretical shear rate, $\dot{\gamma}$, is defined as:

Equation 2.3

$$\dot{\gamma} = \frac{4\Delta p}{\pi r^3 \left(\int_{-x_0}^{x_0} \left(\frac{8\mu}{\pi r^4} \right) dx + \frac{8\mu}{\pi r_0^4} L + R \right)}$$

where r_0 is the nominal radius, μ is the dynamic viscosity, Δp is the pressure differential spanning the region in dx. The equation was developed by assuming that maximum shear rates at the apex may be estimated from a 1-D viscous flow assumptions by applying a series of resistances. Assuming a parabolic velocity profile, the shear rate, $\dot{\gamma}$, is defined as:

Equation 2.4

$$\dot{\gamma} = \frac{4Q}{\pi r^3}$$

where Q is the flow rate. The pressure drop caused from viscous losses was estimated by a force balance. A differential fluid segment can be used with a force balance between shear and pressure integrated along a surface:

Equation 2.5

$$p\pi r^2 - \left(p + \frac{dp}{dx}\Delta x \right)\pi r^2 = \mu\dot{\gamma} 2\pi r \Delta x$$

$$\Rightarrow \Delta p = \int \frac{2\mu\dot{\gamma}}{r} dx = \int \frac{8\mu Q}{\pi r^4} dx$$

where μ is the dynamic viscosity, Δp is the pressure differential spanning the region in Δx. Pressure drops from viscous losses were calculated through Equation 2.5 for the straight vessel section spanning length L (50 vessel diameters) and were computed for the complex geometry of the stenosis region. A summation of pressure loss terms including viscous loss due to the stenosis section, viscous loss due to straight vessel section, and the previously described peripheral resistance are respectively represented as:

Equation 2.6

$$\Delta p = \left(\int_{-x_0}^{x_0} \left(\frac{8\mu}{\pi r^4} \right) dx + \frac{8\mu}{\pi r_0^4} L + R \right) Q$$

The pressure drop equation is a variation of a previous model [28]. The flow rate for a given pressure drop in Equation 2.6 was inserted into Equation 2.4 to estimate shear rates.

A shear-induced platelet activation threshold was considered by [29] based on a curve fit to data collected in [8] with respect to exposure time for constant shear rates. Shear required to activate a platelet for a given time history of a specific shear with a relationship of $\tau^{2.3}t$ results in an approximate constant of 90 dynes$^{2.3}$-s/mm$^{4.6}$. The relationship, integrated along a streamline, can be represented as:

Equation 2.7

$$\tau^{2.3}t = \int_0^T \tau^{2.3} dt = \int_{s1}^{s2} \frac{\tau^{2.3}}{v} ds \qquad \text{(Goodman et al, 2005)}$$

where τ is the shear stress, T is the time interval along a curve, v is the velocity, t is time, s is the distance along a streamline from $s1$ to $s2$, spanning the computational domain. An alternative platelet activation level has also been proposed by [30], using the equation:

Equation 2.8

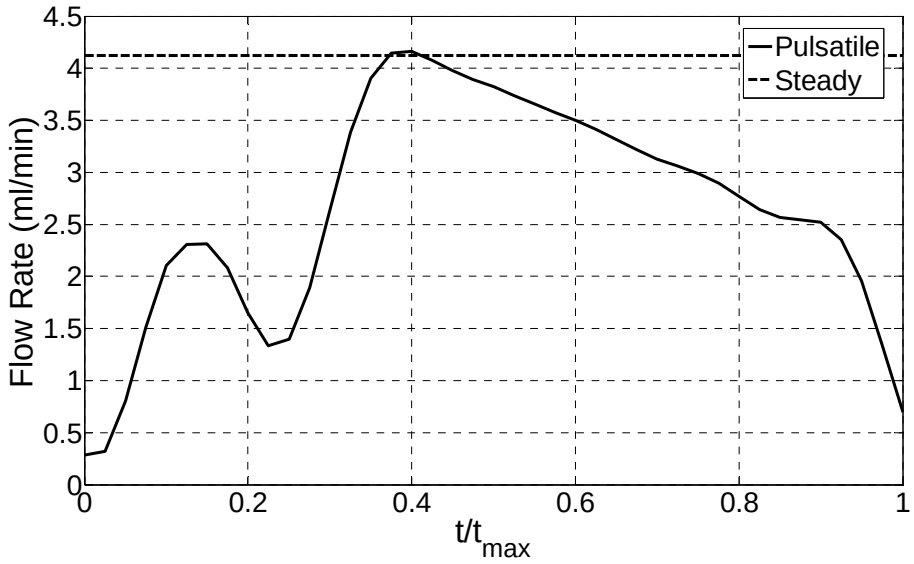
$$\dot{\gamma}t = \int_0^T \dot{\gamma} dt = \int_{s1}^{s2} \frac{\dot{\gamma}}{v} ds \qquad \text{(Bluestein, et al 1997)}$$

Results

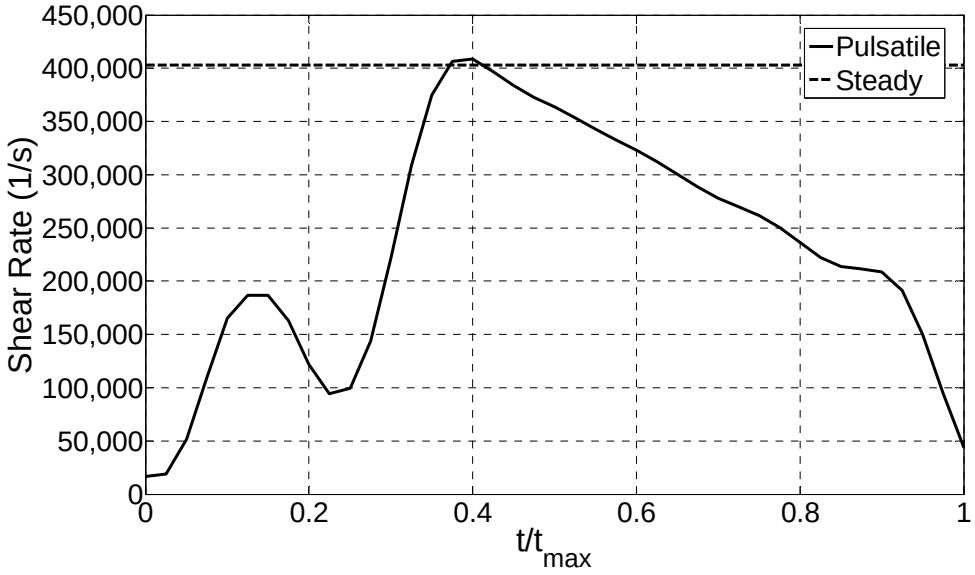
Case 1: Pulsatile flow through a 95% stenosis of a coronary artery.

Pulsatile coronary flow rates ranged from 0 to 4.12 ml/minute, as shown in Figure 2.2A, resulting in time-varying shear rates in the stenosis that reach a maximum of 400,000 s^{-1}, depicted in Figure 2.2B. The hemodynamic solution for a steady flow rate of 4.12 ml/minute (proximal Re=16) yields a shear rate within 2% of the pulsatile solution at the throat of the stenosis. Unsteady effects from pulsatile flow are reduced in a

stenosis region, as expected with a Womersley number of 0.1 at the throat of the 95% stenosis. The Womersley number can characterize unsteady effects and remains lower than 1 for stenoses more severe than a 50% stenosis, indicating that pulsatility can be neglected in the quantification of maximum shear rates in high grade stenosis.



A



B

Figure 2.2: Pulsatile flow through a simulated coronary artery for Case 1 (4 mm – 95% stenosis). A) Input pulsatile flow waveform plotted with a steady flow rate of 4.12 ml/minute. B) Maximum pulsatile shear rates along the stenosis during the flow waveform from A, resulting in a peak shear of 400,000 s^{-1}. The maximum flow solution was predicted well by the steady flow solution. Time is normalized by the length of a cardiac cycle, t_{max}.

Case 2: Hemodynamics in high grade stenoses modeled with a 100 mmHg pressure differential and variable physiologic distal resistance (FFR).

The FFR for Case 2 simulations are given in the curve shown in Figure 2.3, compared with experimental values in the literature for various stenosis degrees. Moderate to high degree stenoses result in a large separation region of reverse flow distal to the throat, as illustrated in Figure 2.4A for a 90% stenosis. The separation point occurs at 125 microns distal to the stenosis apex for a 75% stenosis and moves slightly upstream to 112 microns for a 93-95% stenosis. The centerline velocity reaches approximately 5 m/s for stenoses between 75- 97%.

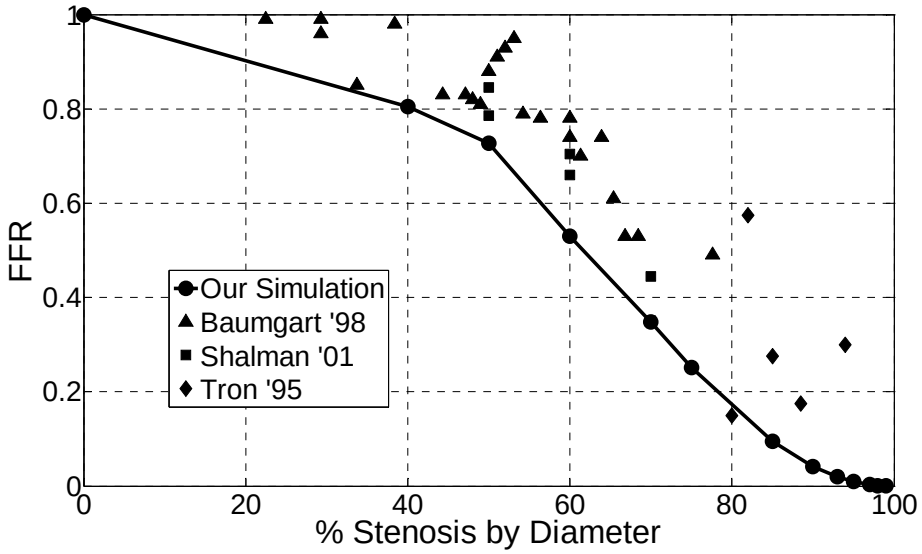
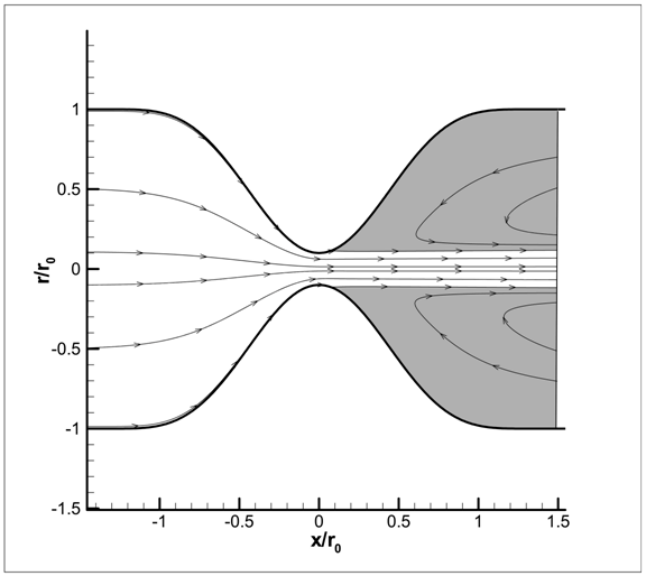


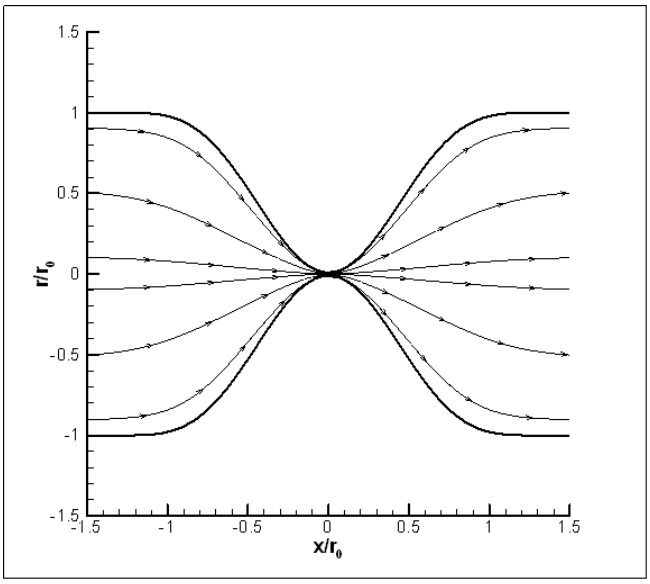
Figure 2.3: Flow characteristics for Case 2 (4 mm – distal resistance – 100 mmHg pressure drop). Computed fractional flow reserve as a function of stenosis degree resulting from simulated distal pressure relative to the inlet pressure, compared with experimental results from the literature [14-16].

New flow behavior emerges as stenoses near occlusion, above 98%. The flow rate decreases such that flow no longer separates. Instead, the flow becomes Stokes-like with a Reynolds number of <1, shown in Figure 2.4B for a 99% stenosis, resulting in the

conversion to a parabolic velocity profile, as shown in Figure 2.5A, as opposed to the blunt profile for a 75-90% stenosis. Figure 2.5B depicts the diminishing reattachment length as the stenosis increases from 75% to 98%.

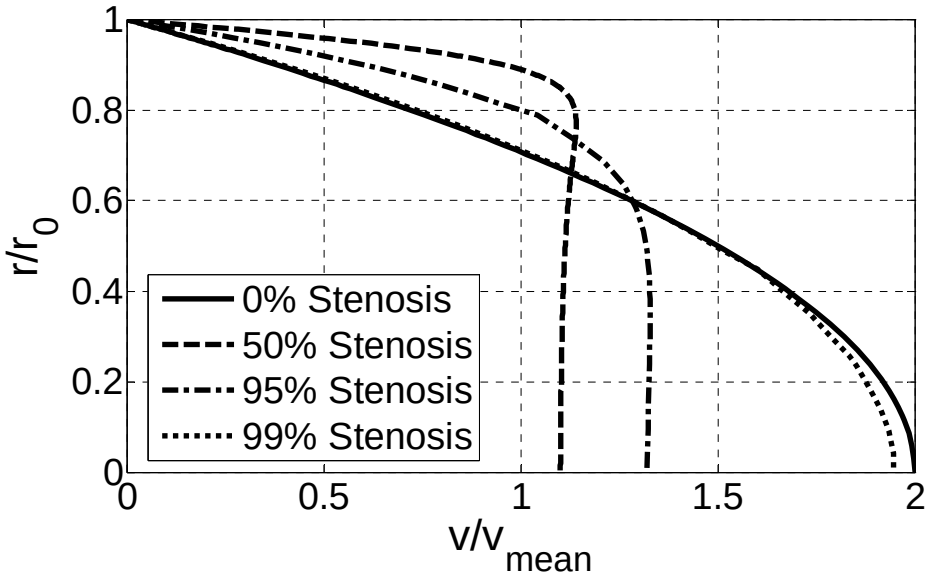


A

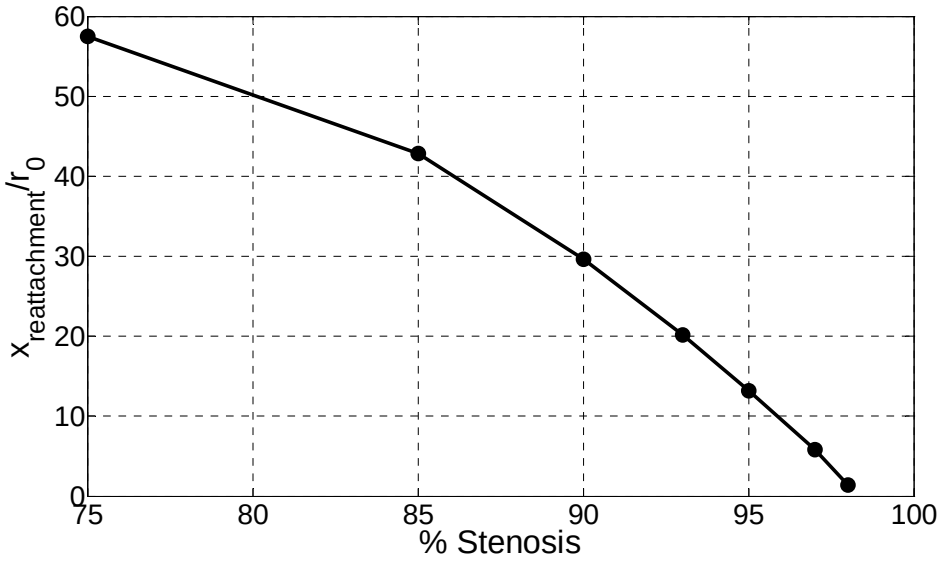


B

Figure 2.4: Flow characteristics for Case 2 (4 mm – distal resistance – 100 mmHg pressure drop). A) Streamlines for a 90% stenosis. B) Streamlines for a 99% stenosis.



A



B

Figure 2.5: Flow characteristics for Case 2 (4 mm – distal resistance – 100 mmHg pressure drop). A) Velocity profiles normalized by the mean velocity plotted relative to the normalized radius based on the nominal vessel radius proximal to the stenosis. Velocity profiles are plotted for no stenosis in addition to a 50, 95, and 99% stenosis. B) Flow reattachment points normalized by the radius of the vessel without the stenosis relative to stenosis degree. Dimensions are normalized by the nominal radius, r_0.

The maximum shear rate increases from a peak shear of 270,000 s^{-1} for a 75% stenosis, to 425,000 s^{-1} for a 98% stenosis, as plotted in Figure 2.6. Wall shear rates increase from the entrance of a stenosis section to a region at or slightly proximal to the apex, with a sharp decline distal to the apex in a region containing separated flow, as shown in the contour plot illustrated in Figure 2.7A with flow entering from the bottom of the figure. The contour plot illustrates that axial shear gradient becomes larger as stenosis degree increases from 75%, resulting in higher maximum shear rates contained within smaller regions of the stenosis.

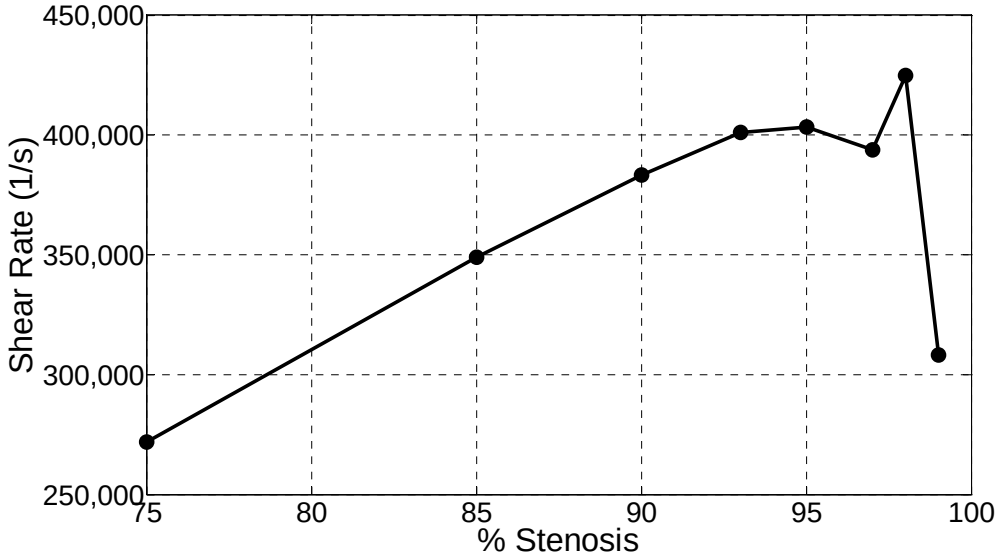
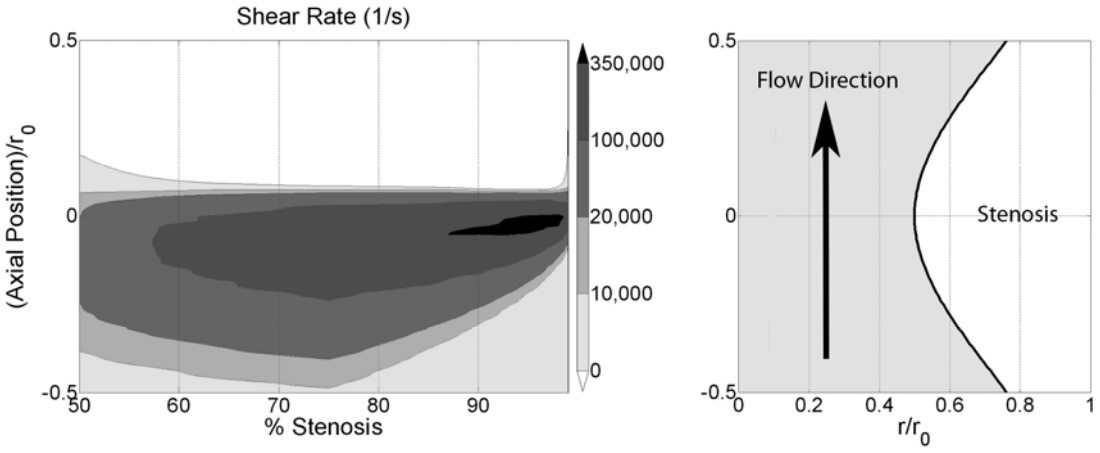


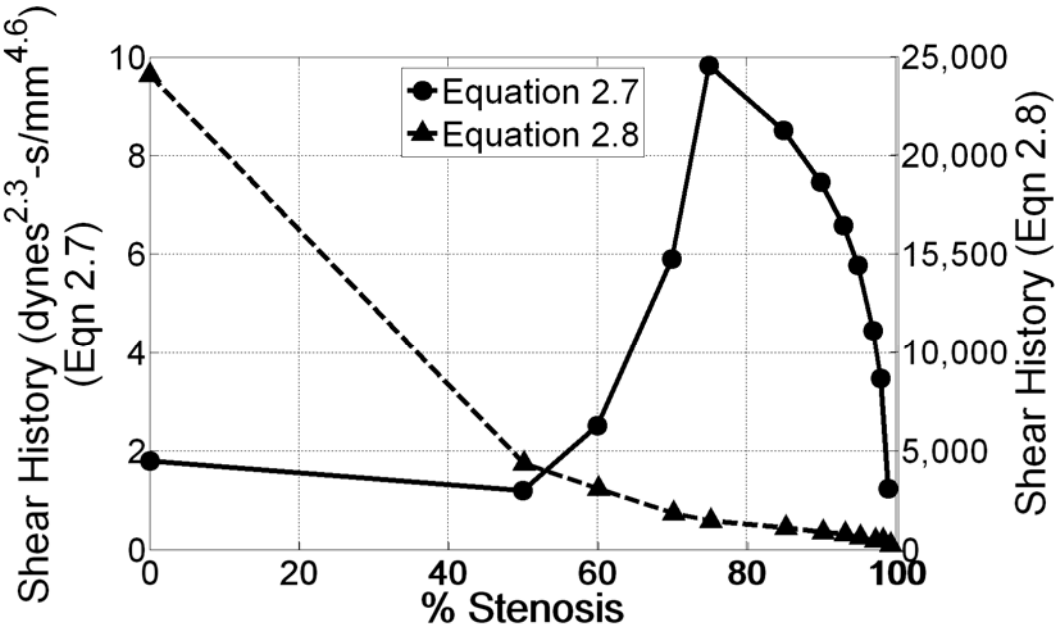
Figure 2.6: Maximum wall shear rates for Case 2 relative to % stenosis

Shear history was evaluated for a streamline passing through a point at 5 microns away from the stenosis apex wall, characterized two ways: i.) Equation 2.7 [29] and ii.) Equation 2.8 [30]. These two calculations for shear history are illustrated in Figure 2.7B for a circulating platelet. Shear history based on Equation 2.7 became maximal with a value of 10 $dynes^{2.3}$-s/$mm^{4.6}$ at a 75% stenosis, then decreased with stenosis degree. Thus

the shear history remains lower than the value of 90 dynes$^{2.3}$-s/mm$^{4.6}$ that is expected to induce shear-dependent activation [8].



A



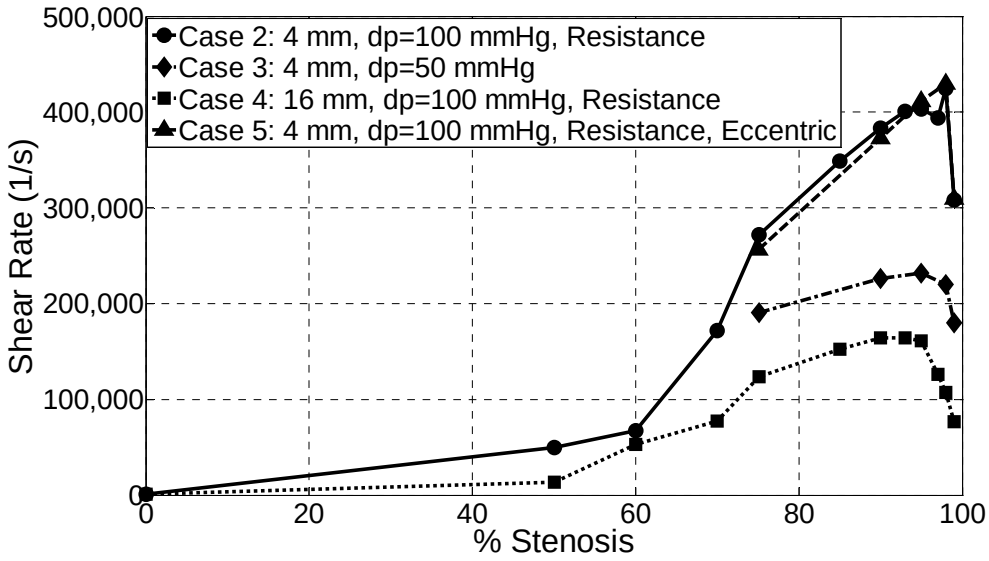
B

Figure 2.7: Shear distribution characteristics along the stenosis for Case 2. A) Contour plot of wall shear rate along the axial wall location of the stenosis (ordinate) at different % stenosis severity (abscissa). The right hand side of the contour plot illustrates a 50% stenosis corresponding to the axial positions of the contour plot for reference. B) Shear history of a particle traveling along a streamline, 5 microns normal to apex of the stenosis at the speed of the fluid. Shear history is plotted relative to Equation 2.7 and Equation 2.8.

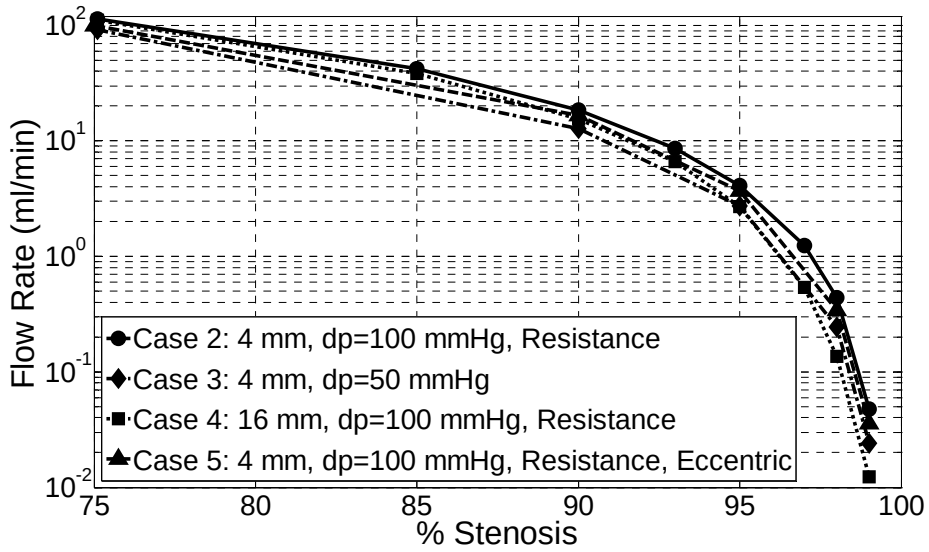
Blood traveling close to the wall convects circulating platelets about to stick to the wall. Flow that is 1.25 microns away from the wall (a platelet's average effective radius [31]) has a particle velocity of 50 cm/s and a shear rate of 425,000 s^{-1} (16,000 dynes/cm^2) for a 98% stenosis. A maximum contact length of a platelet diameter (2.5 μm) at this speed yields a contact time (contact time = contact length/ platelet velocity) of 5 μs for a bond to form. The maximum shear stress of 16,000 dynes/cm^2 would exert 8,000 pN of shear force (force=$\mu * \dot{\gamma} *$ surface area) on the platelet bonds through a force balance just before occlusion for a surface area of 5 μm^2, assuming the platelet is flattened infinitesimally.

Case 3: Effect of a constant 50 mm Hg driving pressure.

A constant pressure drop with no distal resistance, Case 3, can be produced in vitro and provides an alternative flow condition, while providing pressure drops that result in low shear rates relative to Case 2. Figure 2.8A illustrates the reduction in shear rate of about 30-50% for Case 3 compared to Case 2, corresponding to a decrease in flow rate by 20- 50%, as illustrated in Figure 2.8B. Nonetheless, shear rates for Case 3 reached 200,000 s^{-1} for severe stenoses > 75%.



A



B

Figure 2.8: Effect of outlet resistance and stenosis length on the maximum shear in a stenosis. A) Maximum shear rates and B) Flow rates for Case 2 (4 mm – distal resistance – 100 mmHg pressure drop), Case 3 (4 mm – no distal resistance – 50 mmHg pressure drop), Case 4 (16 mm – distal resistance – 100 mmHg pressure drop), and Case 5 (4 mm – distal resistance – 100 mmHg pressure drop - eccentric). Note that flow rates for stenoses <75% are 120 ml/min.

Case 4: Effect of stenosis length.

Atherosclerotic plaque can vary in length. An increase in the stenosis length by a factor of 4 to 16 mm is defined as Case 4. The extended length reduces the shear rate by a factor of 4 for a 98% stenosis (Figure 2.8A), in comparison with Case 2, corresponding to a factor of 3.25 decrease in flow rate for the elongated stenosis (Figure 2.8B). The decrease in flow rate partially explains why the shear rates decrease for Case 3 and 4 relative to Case 2. Maximum shear rates for the extended length severe stenoses were near 125,000 s^{-1}. The maximum shear rate for the 16 mm long stenosis occurs at 93% reduction by diameter, a slightly lower degree of stenosis than for the 4mm long stenosis.

Case 5: Effect of eccentricity.

Plaque does not typically grow in the idealized axisymmetric shapes for Cases 1-4. Therefore, plaque eccentricity was fully modeled in 3-D. The resultant maximum shear rates were within 6% of axisymmetric 4mm long stenoses hemodynamic shear values, as depicted in Figure 2.8A. The flow rate (Figure 2.8B) was within 12 % of case 2. Thus, for severe stenoses, eccentricity is not a major factor for local hemodynamics.

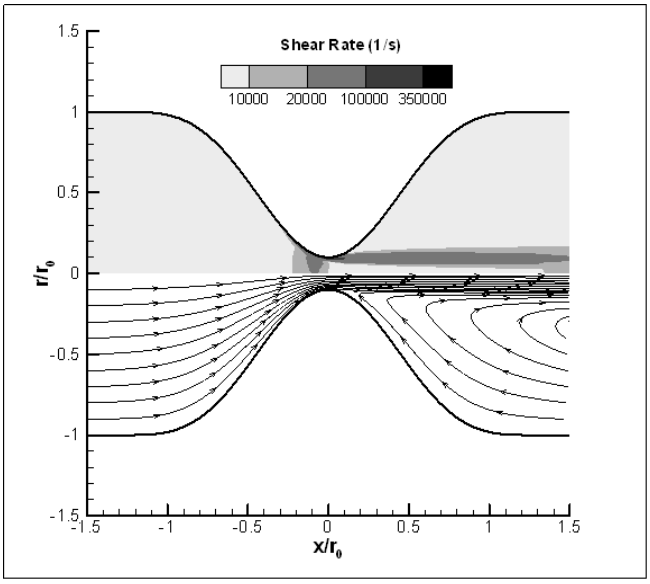
Case 6: Effect of thrombus surface roughness.

A growing thrombus may not have the smooth, hourglass surface of the stenoses in Cases 1-5. Case 6 simulates a rough thrombus surface on a 4 mm long stenosis. The roughness is built around the smooth 90% stenosis, resulting in 200 μm hills and valleys, which can increase the local stenosis to 97%.

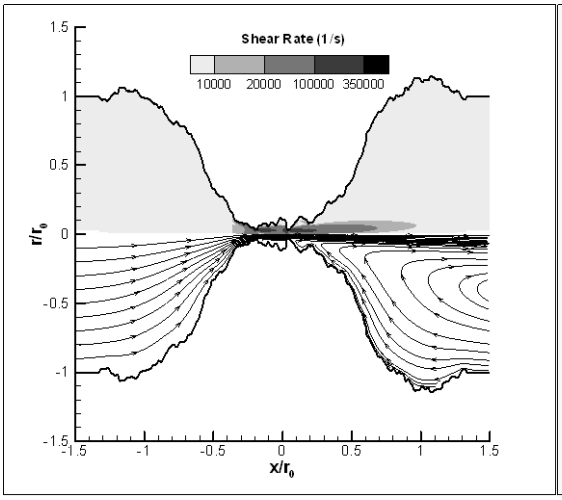
For a smooth 90% stenosis (Case 2) (Figure 2.9A), the shear rate reaches a maximum of about 383,000 s^{-1}. In contrast, the rough surface is shown in Figure 2.9B with a magnified section in Figure 2.9C illustrating the shear rate at the throat. Shear rate peaks develop along the hills with small recirculation eddies in valleys of the rough surface, causing platelets to experience varying shear stresses near the apex of a stenosis. The irregular thrombus surface results in focal areas of increased shear rate relative to the smoothly varying surface. Maximum shear rate reached 610,000 s^{-1} for rough surfaces. This is higher than the shear rate of 383,000 s^{-1} for a smooth 90% stenosis or the 394,000 s^{-1} for a smooth 97% stenosis. The 59% increase in focal shear (90% stenosis) from 200 micron rough features can be contrasted with the 75% reduction in shear (99% stenosis) created for elongated plaques of 16 mm as shown in Table 2.2. The maximum shear in a stenosis is then sensitive to the additional length scales of roughness and plaque length.

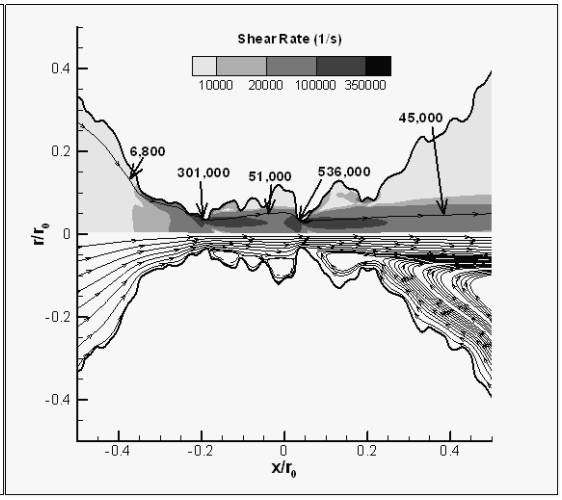
Table 2.2: Shear Rates as a function of feature length scales, including stenosis length and rough surface length scales for a 97% stenosis

Feature Length Scale	Shear Rate	Case
16,000 μm	**130,000 s^{-1}**	**4**
4,000 μm	**390,000 s^{-1}**	**2**
140 μm	**610,000 s^{-1}**	**6**



A





B C

Figure 2.9: Effect of surface roughness on local shear rates for a 90% nominal stenosis. A) Shear rate contours and streamlines (smooth surface Case 2). B) Shear rate contours and streamlines (rough surface Case 6). Magnified shear rate contours and streamlines with a near wall streamline marked with shear rates at various points (rough surface Case 6). Note that for (B), the nominal average stenosis is 90% by diameter, but the rough surface protrudes to a 97% stenosis at the peaks.

Simplified Estimates of Shear

Shear rates in tubes may be estimated using simplified 1-D assumptions of flow controlled by viscous losses modeled in Equation 2.3. Shear in Equation 2.3 is plotted relative to numerically determined shear rates of Case 2 and 4 in Figure 2.10. Equation 2.3 does not account for the blunt velocity profile or hydraulic losses caused by phenomena such as flow separation, causing poor agreement with numerical results in the 75-85% stenosis range. Predictions of wall shear from Equation 2.3 become accurate above 98% for a 4 mm long stenosis, or above 95% for a 16 mm long stenosis, as the viscous effects dominate and separation disappears.

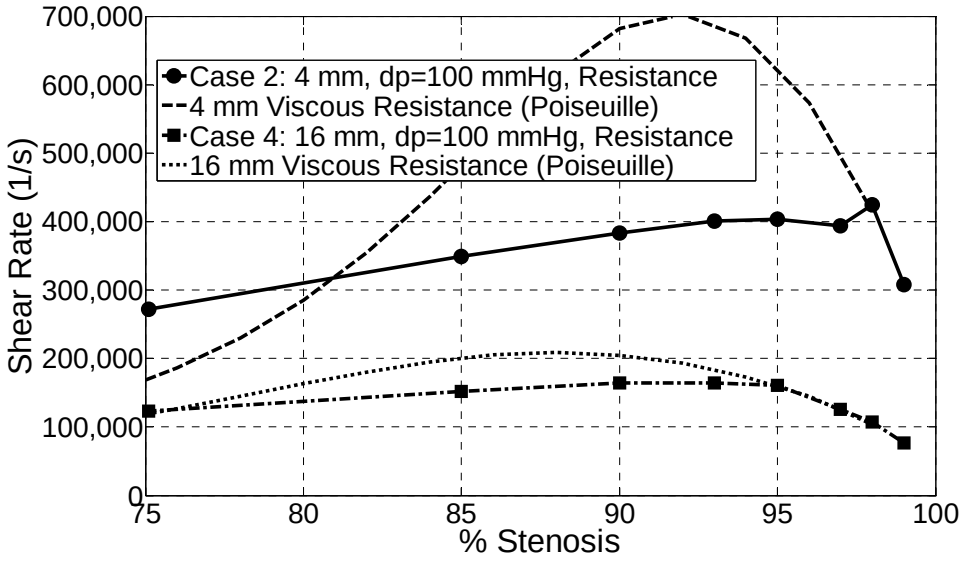


Figure 2.10: Shear rates through numerical predictions for Case 2 (4 mm – distal resistance – 100 mmHg pressure drop) and Case 4 (16 mm – distal resistance – 100 mmHg pressure drop) approach shear predicted from Poiseuille flow assuming only viscous energy losses.

Discussion

The fluid mechanic results quantify the extremely high blood velocities (5 m/s) and wall shear rates (>250,000 s^{-1}) that develop within severe arterial stenoses (≥75%). Previous research has focused on stenoses of low to moderate severity, which results in lower shear rates. Shear rates for the severe coronary stenoses result in local peak values that may reach 610,000 s^{-1}, which is 7 times greater than the maximum shear (84,000 s^{-1}) previously described for a moderate stenosis of 65% [18].

Atherothrombosis that forms near the throat of the stenosis is subjected to very high hemodynamic shear forces. The hemodynamic force (8000 pN) applied to a platelet attached to the surface is 2 orders of magnitude greater than the reported optimal bond strength (75 pN for longest bond lifetime [11]) between platelet GPIbα and the vWF A1 domain. The hemodynamic force estimate is conservative compared to Stokes flow approximations through [32]. The biophysical requirements for thrombus growth at this shear are an indication that platelet bonding may be different from the single GPIbα-A1 bonds generally hypothesized for high shear attachment. Multiple bonds may be required or different binding mechanisms may be involved, similar to the transition from fibrinogen binding to vWF binding near 1,000 s^{-1} [9]. Coiled vWF transitions to uncoiled vWF near shear rates of 3,000-8,000 s^{-1} [33]. So it comes into question what platelet binding mechanisms apply at shear rates of 200,000 s^{-1}.

The capture of new blood platelets onto the mural thrombus also challenges physical bonding kinetics. As the moving platelet passes a point at rapid speed, it must be captured by a bond forming in a very short time for bond formation (5 μs).

Circulating platelets traversing the length of the stenosis are not expected to experience shear-dependent activation based on shear history calculations. Thus, shear activation of the convecting platelets does not appear to be present for initial platelet attachment in stenoses. Platelet attachment in severe stenoses occurs under the adverse binding conditions of short residence time, while resisting high shear forces, without prior shear activation.

Activation based on shear history was calculated through Equation 2.7 and Equation 2.8. Results illustrate that values based on Equation 2.7 reach values that are on the order of Hellums values [8]. However, values from Equation 2.8 reach near 3,000 for a 60% stenosis. Results for a similar stenosis have been previously reported to reach approximately 570 in work by Bluestein [30]. Equation 2.8 that was developed by Bluestein essentially scales with axial distance of a streamline relative to the distance from the wall, explaining the discrepancy in the data if Bluestein measures shear history over a streamline located at a different radial location than us. The reason the equation scales in such a way is that the near wall speed is approximately the shear rate multiplied with the distance from the wall. Therefore, the transit time is the axial length of the streamline divided by this value, leaving a inverse relationship with shear. Bluestein's equation then multiplies the transit time with shear, resulting in a quantity that is approximately equal to the axial distance of the streamline divided by streamline's distance from the wall.

Our calculations for stenoses <75% are not presented in detail, but were similar to those found in the literature. Maximum shear rates for our simulation of a 16 mm long 68 % stenosis results in 72,000 s^{-1} and 53,000 s^{-1} for a 60% stenosis. Calculations from

[17] result in a maximum wall shear rate of 72,000 s^{-1} for a 9 mm long 68% stenosis by diameter and [18] found 44,000 s^{-1} for a 58% stenosis. Minor discrepancies between the literature and our results are likely due to differences in stenosis geometries.

We further show that unsteady effects from the cardiac cycle have minimal impact on maximum shear rates for severe stenoses, consistent with the low Womersley number. Similarly, eccentricity had little effect on flow rates or wall shear rates for severe stenoses, which matches reported results for a blunt plug creating severe stenoses [22]. Only smoothly varying eccentricity was considered in our study. Eccentric stenoses of lower severity may have more energy loss than concentric stenoses [21]. Vessel curvature proximal or distal to a stenosis may also result in more significant differences in the recirculation region size and wall shear rate distribution relative to the idealized straight vessel cases considered in our study [20].

A limitation in our study is that we have not included Fahraeus-Lindquist effects [26]. We assumed idealized geometries and rigid walls typical of a calcific stenosis, thereby eliminating the possibility for the vessel to expand or collapse. The local hemodynamics may be further affected by non-rigid stenoses that may potentially collapse from low Bernoulli pressures near the stenosis apex [34-35].

The hemodynamics of arteries nearing thrombotic occlusion leads to high forces and fast kinetics of bonding within the severe stenosis. Platelet adherence and accumulation must occur in the face of pathological high shear stresses exceeding 10,000 dynes/cm^2, creating rapid bonds in less than 5 μs, with bond strengths up to 8000 pN without prior activation of circulating platelets. Future studies at pathologic high shear

rates described may reveal processes and mechanisms that differ from thrombus formation at lower shear rates.

References

1. Yutani, C., et al., *Coronary atherosclerosis and interventions: pathological sequences and restenosis.* Pathology International, 1999. **49**(4): p. 273-290.
2. Hathcock, J.J., *Flow Effects on Coagulation and Thrombosis.* Arteriosclerosis, Thrombosis, And Vascular Biology, 2006. **26**(8): p. 1729-1737.
3. Sakariassen, K.S., et al., *Collagen type III induced ex vivo thrombogenesis in humans. Role of platelets and leukocytes in deposition of fibrin.* Arteriosclerosis, Thrombosis, And Vascular Biology, 1990. **10**(2): p. 276-284.
4. Markou, C.P., et al. *Role of high wall shear rate on thrombus formation in stenoses*. in *Advances in Bioengineering*. 1993. New Orleans: ASME.
5. Ku, D.N. and C.J. Flannery, *Development of a flow-through system to create occluding thrombus.* Biorheology, 2007. **44**(4): p. 273-284.
6. Badimon, L. and J.J. Badimon, *Mechanisms of arterial thrombosis in nonparallel streamlines: platelet thrombi grow on the apex of stenotic severely injured vessel wall. Experimental study in the pig model.* The Journal Of Clinical Investigation, 1989. **84**(4): p. 1134-1144.
7. Zydney, A.L. and C.K. Colton, *Augmented solute transport in the shear flow of a concentrated suspension.* PhysicoChemical Hydrodynamics, 1988. **10**(1): p. 77-96.
8. Hellums, J.D., *1993 Whitaker Lecture: biorheology in thrombosis research.* Annals Of Biomedical Engineering, 1994. **22**(5): p. 445-455.
9. Savage, B., E. Saldívar, and Z.M. Ruggeri, *Initiation of platelet adhesion by arrest onto fibrinogen or translocation on von Willebrand factor.* Cell., 1996. **84**(2): p. 289-297.
10. Ruggeri, Z.M., et al., *Activation-independent platelet adhesion and aggregation under elevated shear stress.* Blood, 2006. **108**(6): p. 1903-1910.
11. Yago, T., et al., *Platelet glycoprotein Ibalpha forms catch bonds with human WT vWF but not with type 2B von Willebrand disease vWF.* The Journal Of Clinical Investigation, 2008. **118**(9): p. 3195-3207.
12. Spaan, J.A.E., et al., *Physiological basis of clinically used coronary hemodynamic indices.* Circulation, 2006. **113**(3): p. 446-455.
13. Gould, K.L., K. Lipscomb, and G.W. Hamilton, *Physiologic basis for assessing critical coronary stenosis. Instantaneous flow response and regional distribution during coronary hyperemia as measures of coronary flow reserve.* The American Journal Of Cardiology, 1974. **33**(1): p. 87-94.
14. Baumgart, D., et al., *Improved assessment of coronary stenosis severity using the relative flow velocity reserve.* Circulation, 1998. **98**(1): p. 40-46.
15. Shalman, E., et al., *Pressure-based simultaneous CFR and FFR measurements: understanding the physiology of a stenosed vessel.* Computers In Biology And Medicine, 2001. **31**(5): p. 353-363.
16. Tron, C., et al., *Comparison of pressure-derived fractional flow reserve with poststenotic coronary flow velocity reserve for prediction of stress myocardial perfusion imaging results.* American Heart Journal, 1995. **130**(4): p. 723-733.

17. Siegel, J.M., et al., *A scaling law for wall shear rate through an arterial stenosis.* Journal of Biomechanical Engineering, 1994. **116**(4): p. 446-451.
18. Strony, J., et al., *Analysis of shear stress and hemodynamic factors in a model of coronary artery stenosis and thrombosis.* American journal of physiology. Heart and circulatory physiology, 1993. **265**(5): p. H1787-1796.
19. Tangelder, G.J., et al., *Wall shear rate in arterioles in vivo: least estimates from platelet velocity profiles.* The American Journal Of Physiology, 1988. **254**(6 Pt 2): p. H1059-H1064.
20. Steinman, D., et al., *Flow Patterns at the Stenosed Carotid Bifurcation: Effect of Concentric versus Eccentric Stenosis.* Annals Of Biomedical Engineering, 2000. **28**(4): p. 415-423.
21. Young, D.F. and F.Y. Tsai, *Flow characteristics in models of arterial stenoses. I. Steady flow.* Journal of Biomechanics, 1973. **6**(4): p. 395-410.
22. Seeley, B.D. and D.F. Young, *Effect of geometry on pressure losses across models of arterial stenoses.* Journal of Biomechanics, 1976. **9**(7): p. 439-446, IN3, 447-448.
23. Deshpande, M.D., *Steady laminar and turbulent flow through vascular stenosis models*. 1977, Georgia Institute of Technology: Atlanta.
24. Ku, D.N., *Blood flow in arteries*. 1997. p. 399.
25. Wootton, D.M., et al., *A mechanistic model of acute platelet accumulation in thrombogenic stenoses.* Annals of biomedical engineering., 2001. **29**(4): p. 321-329.
26. Gaehtgens, P., *Flow of blood through narrow capillaries: rheological mechanisms determining capillary hematocrit and apparent viscosity.* Biorheology, 1980. **17**(1-2): p. 183-189.
27. Berne, R.M. and M.N. Levy, *Principles of Physiology*. 3rd ed. 2000, St. Louis: Mosby.
28. Mates, R.E., et al., *Fluid dynamics of coronary artery stenosis.* Circulation Research, 1978. **42**(1): p. 152-162.
29. Goodman, P.D., et al., *Computational model of device-induced thrombosis and thromboembolism.* Annals Of Biomedical Engineering, 2005. **33**(6): p. 780-797.
30. Bluestein, D., et al., *Fluid mechanics of arterial stenosis: relationship to the development of mural thrombus.* Annals Of Biomedical Engineering, 1997. **25**(2): p. 344-356.
31. Fung, Y.C., *Biomechanics : Mechanical Properties of Living Tissues*. 1981, New York :: Springer-Verlag.
32. Goldman, A.J., R.G. Cox, and H. Brenner, *Slow viscous motion of a sphere parallel to a plane wall--II Couette flow.* Chemical Engineering Science, 1967. **22**(4): p. 653-660.
33. Schneider, S.W., et al., *Shear-induced unfolding triggers adhesion of von Willebrand factor fibers.* Proceedings Of The National Academy Of Sciences Of The United States Of America, 2007. **104**(19): p. 7899-7903.
34. Tang, D., et al., *Simulating cyclic artery compression using a 3D unsteady model with fluid-structure interactions.* Computers & Structures, 2002. **80**(20-21): p. 1651-1665.

35. Downing, J.M. and D.N. Ku, *Effects of frictional losses and pulsatile flow on the collapse of stenotic arteries.* Journal Of Biomechanical Engineering, 1997. **119**(3): p. 317-324.

CHAPTER 3. SPECIFIC AIM 2: EFFECT OF HIGH WALL SHEAR ON *IN VITRO* THROMBUS GROWTH OVER A STENOSIS

Introduction

Arteries narrowed by atherosclerotic plaque can result in unstable angina, stroke, myocardial infarction, and sudden cardiac death. Atherosclerosis develops over decades, creating a stenosis, which is a hemodynamic obstruction in the blood flow, but the sudden onset of thrombosis results in thrombi that superimpose the initial lesion, restricting the blood vessel at much faster time scales [1-2]. Atherothrombosis, or platelet aggregation, on or near an atheroma is found in 90% of the cases of acute myocardial infarction (AMI) through postmortem studies [3-6]. More specifically, thrombus growth can be found, in flow studies, at the apical region of a stenosis, the narrowest section of the lumen [7-11]. Hemodynamics are hypothesized to play a role in causing growth at this location, since a region of locally high shear reaching as high as 80,000 s^{-1} [12-14] can exist on the surface of the stenosis, as opposed to the surrounding normal vessel with shear rates of 400 s^{-1}.

Previous studies of atherothrombosis have provided clues about the role of wall shear rates in effecting thrombus growth. Postmortem studies have found that atherothrombosis is platelet rich in high shear regions [15], congruent with findings from *ex vivo* baboon studies [16]. Bulk thrombus growth in *ex vivo* baboon shunts have shown an increasing trend between platelet deposition rate and shear rates [10, 17] until 7,500 s^{-1}, with similar results for porcine blood [18]. Higher platelet deposition rates were found for shear rates of 10,000 s^{-1} [19-25]. Controversy remains for shear rates reaching 32,000 s^{-1}. One report shows even greater growth rates at this shear [2], while another group

reports a decreased growth rate [19-20]. These experiments were limited because they only observed thrombus growth at a few discrete shear rates. Furthermore, these studies only consider initial shear rates, while neglecting dynamic changes in shear caused by decreasing lumen size through thrombus growth. However, large changes in shear can occur from even small changes in the lumen size or shape for severe stenoses [12].

To evaluate the role of wall shear rates during thrombus growth, we compute the hemodynamics around growing thrombi through computational fluid dynamics (CFD). While covering a large range of shear, 0-100,000 s^{-1}, we implement an *in vitro* model of thrombus growth by perfusing blood over an atheroma-shaped collagen coated stenosis. CFD is utilized as the geometry does not lend itself to simple Poiseuille solutions of the flow field. Steady flow is used, since pulsatility has only secondary effects on the shear and mass transport over a stenosis apex in the coronary artery [26-28]. In contrast to previous studies, we report on the continuous variation of thrombus growth rates relative to temporally changing shear rates.

Methods

A. Experimental Apparatus

Thrombosis was induced through a variation of the single pass flow system previously described by Ku and Flannery [2]. The original system controlled flow through a constant pressure head. This study, instead, utilizes a system where flow was controlled through a syringe pump, providing a relatively constant volumetric flow rate. The constant flow rate allows us to develop well posed boundary conditions for CFD. A single pass system avoids the risk of recirculating activated platelets or platelet-depleted blood. Porcine blood was collected from an abattoir (Holifield Farms, Covington, GA)

and was anticoagulated with 3.5IU/100ml heparin (Sigma Aldrich, St. Louis, MO) to prevent clotting while be transported to the experimental site. Experiments were performed within 10 hours of the blood collection. A Coulter counter was used to quantify the average mean platelet volume, average platelet concentration, and average hematocrit.

Whole blood was perfused through an axisymmetric stenosis fabricated by a professional glass blower using a Pyrex® capillary tube. The stenosis geometry is intended to mimic the shape of an atherosclerotic lesion. The test section consists of a nominal diameter of 1.5 ± 0.1 mm with a smoothly varying sinusoidal 74-84% stenosis by diameter. The geometry of the stenosis is approximately axisymmetric and axially spans 4-6 mm in the central portion of the capillary. Axisymmetry simplifies the analysis of thrombus growth and hemodynamic evaluation through CFD. Table 3.1 lists the stenosis severity and corresponding flow rates for the 5 cases of thrombus growth studied here.

Table 3.1: Experimental conditions of the present study

Case	% Stenosis by Diameter	Flow Rate (ml/min)
1	84	0.25
2	82	0.42
3	85	0.28
4	81	0.51
5	80	0.54

Flow rates ranged from 0.25 to 0.54 ml/min, resulting in an upstream Reynolds number (Re) that ranged from 0.95 to 2.1. A region for boundary layer development was included by extending straight tubing 20 diameters upstream and downstream of the

stenosis section, a length determined to be sufficient for flow development. The theoretical entrance length for fully developed flow (0.06 Re*D) is only ~ 0.2 mm. Near wall platelet concentrations have been shown to experimentally develop approximately within 25 diameters for a 40% hematocrit at a Re of 2 [29]. Silicone tubing upstream of the stenosis combined with the entrance region of the test section should allow for a fully developed platelet concentration profile.

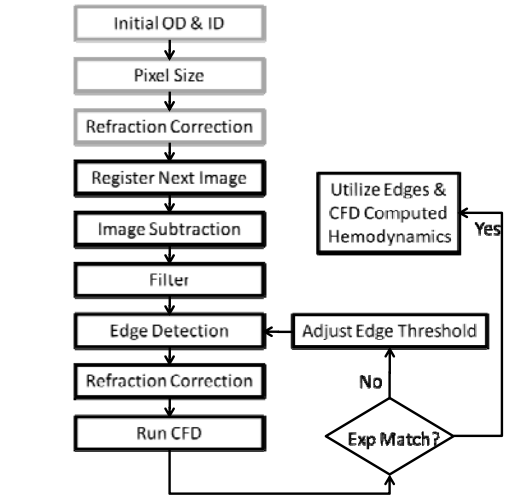
To create an adhesive surface for thrombus to form, the capillary tubes were coated with fibrillar collagen type I (1 mg/ml, Sigma Aldrich, St. Louis, MO), which is a thrombogenic protein. Collagen type I is also physiologically relevant because it is likely exposed upon plaque rupture or erosion. The stenosis, or region of capillary curvature, was uniformly coated, extending downstream of the stenosis until the end of the capillary. Correspondingly, no collagen was present upstream of the stenosis.

To verify proper pressure drop predictions from CFD, pressure was monitored by placing a pressure transducer (Harvard Apparatus, South Natick, Massachusetts) immediately upstream to the capillary. The downstream end of the test section was exposed to atmosphere in all experiments. Silicone tubing was used to connect the components of the experimental apparatus. Experiments were terminated with occlusion defined when the upstream pressure reached 50 mmHg.

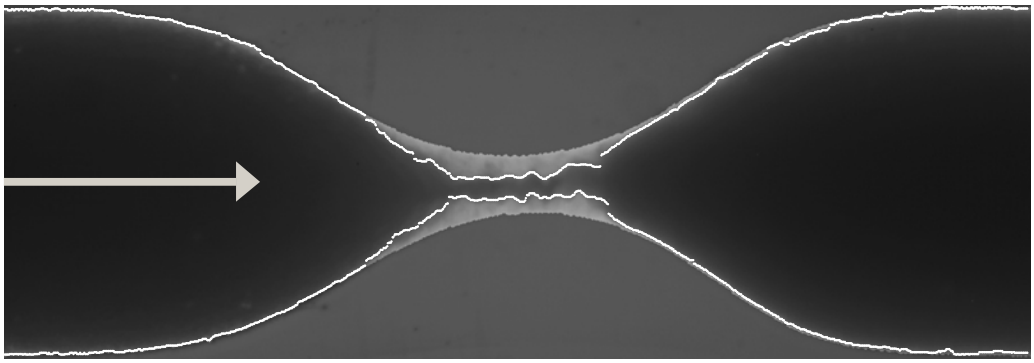
B. Growth quantification

Thrombus thickness is measured once every second using light microscopy. Whole blood can be distinguished from thrombus or glass because less light is transmitted through whole blood. Thrombus formation was imaged with a PIXELFLY QE (PCO Kelheim, Germany) camera through an inverted microscope (Zeiss Stemi 2000-c,

Oberkochen, Germany). The optical resolution of the system is approximately 10 microns. A halogen lamp was used with a fiber optic backlight source, providing constant incident intensity. The images were post-processed using Matlab (ver 7.0.1.24704 R14 Service Pack 1; The Mathworks, Inc., Natick, MA). The method utilized for determining the initial stenosis edge and subsequent thrombus edge is diagramed in Figure 3.1A.



A



B

Figure 3.1: A) Schematic of the edge detection technique used in this study. B) Edge detection for Case 1 after 7 minutes of blood perfusion, prior to refraction correction. The initial stenosis is 85% by diameter and the final stenosis after thrombus growth is 95% by diameter. The darkened outer region corresponds to the initial stenosis, while the inner lines correspond to the detected edges. Blood can be seen as the dark substance in the center of the image.

To determine the glass wall of the test section, the pixels in the initial blood perfusion image were first averaged with the first 5 images temporally. Averaging aided in filtering out small variations between images, such as moving platelet aggregates on the stenosis surface. Subsequently, the initial lumen edge was determined through the Sobel gradient method using the Matlab *edge* function. To determine the perceived lumen radius at a given axial location, the number of pixels spanning from edge to edge were counted. The pixel physical dimension was determined by the number of pixels spanning the known outer diameter of the capillary (#pixels/(2 R_o)). Therefore, the perceived inner radius, *y*, of the vessel is known. However, the cylindrical nature of the capillary creates refraction in the image, which was corrected through Snell's law.

To determine the thrombus edge, subsequent snapshots were averaged with the nearest 5 images. Images were registered and the initial non-thrombosed snapshot was subtracted from subsequent images to filter out features that are consistent through all images, thus providing a cleaner picture for edge detection. After applying a Gaussian filter to smooth out sharp changes in the subtracted image, we defined a perceived thrombus edge based on a light intensity threshold technique. Any transmitted light below a threshold value was considered to be whole blood. The transition from whole blood to thrombus or glass was considered an edge. Refraction was corrected for through Snell's law. To illustrate the edge detection technique, an example thrombus boundary measured prior to refraction correction is illustrated in Figure 3.1B. Refraction correction would radially offset the edges, such that they no longer align with the perceived image. The optical refraction led to corrections of up to 45%. The corresponding initial stenosis prior to refraction correction is outlined in a shade of dark gray.

C. Refraction Correction

A conversion from the perceived stenosis radius to the actual stenosis radius, y', was determined by considering light transmitted through air and the glass capillary:

Equation 3.1

$$y' = \left(y - R_o \cos(\alpha) \tan(\alpha - \beta) \right)$$

$$\alpha = \arcsin\left(\frac{y}{R_o} \right)$$

$$\beta = \arcsin\left(\frac{n_{air} y}{n_{pyrex} R_o} \right)$$

where y is the perceived inner radius of the vessel, R_o is the outer physical radius of the vessel, and n_{air} is the refractive index of air, taken as 1. The refractive index of the stenosis, n_{pyrex}, which was created by heating glass, was determined by solving for n_{pyrex} in Equation 3.1, assuming that y' was the non-stenosed inner radius of the vessel (750 μm). The refractive index was calculated for all 5 stenoses and yielded an average of 1.45.

Refraction of thrombus was corrected through Snell's law, similar to Equation 3.1, except temporally later edges are observed through an additional layer of thrombus. Therefore, the updated equation for refraction correction of a thrombus edge was determined by solving:

Equation 3.2

$$y' = R_i \sin(\alpha - \beta + \phi) - R_i \cos(\alpha - \beta + \phi) \tan(\alpha - \beta + \phi - \theta)$$

$$\phi = \arcsin\left(\frac{n_{air} y}{n_{pyrex} R_i}\right)$$

$$\theta = \arcsin\left(\frac{n_{air} y}{n_{thrombosis} R_i}\right)$$

where R_i is the initial lumen radius of the vessel determined through y' of Equation 3.1. The refractive index of thrombosis was taken as 1.4 based on the refractive index of platelets [30].

D. CFD: Hemodynamic quantification of instantaneous shear rate

Hemodynamic conditions change as thrombosis proceeds because thrombi superimpose the initial stenosis, which effectively changes the wall boundary for the flow of blood. Using the thrombus edge as a wall boundary for CFD and experimental flow rates as an inlet condition, we obtain computed results of the flow field every 10 seconds. 10 seconds was chosen because thrombi exhibited sufficient growth to vary the mean wall shear rate relative to a previous time step by up to 20%. To create a CFD solution of the flow field, a mesh was developed in the region of blood flow based on the thrombus and stenosis boundary, using Gambit 2.4.6 (ANSYS Santa Clara, CA). A structured mesh was created in the boundary layer, transitioning into an unstructured triangular mesh in the stenosis region near the core of the vessel. An unstructured quadrangle mesh was placed upstream and downstream to the stenosis. The mesh resolution was increased until a mesh independent solution was obtained.

To compute the flow field, the incompressible Navier-Stokes equations were solved using a second order upwind scheme in the finite-volume based commercial computational fluid dynamics (CFD) package, Fluent 6.3.26 (ANSYS Santa Clara, CA). SIMPLE velocity-pressure coupling was used. The solution was assumed to converge when continuity and velocity component residuals dropped below 10^{-10}. Convergence was also verified with a mass balance between the inlet and outlet. Two-dimensional axisymmetry was assumed with a lumen diameter corresponding to the distance between detected thrombus or stenosis edges. The flow in the CFD solution was assumed to be laminar because upstream Re in our system remained below 2, which is well below the turbulence transition upstream Re for a stenosis [26]. Blood was considered Newtonian because shear rates in the flow field were sufficiently high to break up rouleux, thus minimizing non-Newtonian phenomena. Thrombus growth has the potential to create unsteady flow if the boundary changes at a high enough speed. However, thrombus growth rates found in this study generally do not exceed 0.5 μm/s, which can be compared with average inlet blood velocities of 5,000 μm/s. Therefore, unsteady effects from thrombus growth are negligible relative to convection in the current study.

E. Correction of thrombus edge detection by pressure drop

The edges of thrombi are not distinct in optical images, similar to many machine vision applications. Therefore, our method of edge detection was verified to obtain the most accurate edge choices within the limitations of the techniques used in this study. To verify our edge choice, the pressure drop determined through computational evaluation of flow through the stenosis and thrombi was compared with the experimental pressure drop across the same region. The pressure drop is a sensitive parameter for verifying a radius

because simple Hagen-Poiseuille flow assumptions and blunt plug stenoses relate the pressure drop to a constant flow rate by the inverse of the radius to the fourth power [26]. The optical intensity threshold used for edge detection was adjusted in cases where the computed pressure drop varied by more than 30% of the measured pressure drop, which would correspond to a radius offset of only 2.3% based on the fourth power assumption. The process of detecting an edge was reiterated until pressure drops were within the specified bounds based on the updated intensity thresholds. This edge correction method created changes in thickness averaging up to 25% relative to edge estimations by eye.

F. 3-D vs. 2-D hemodynamics

Variations in the flow field caused by stenosis eccentricity and out-of-plane thrombus growth may vary from the idealized two-dimensional axisymmetric geometries assumed in the present study. Therefore, we also studied flow through a three-dimensional thrombus boundary in one case to verify that wall shear rates determined through two-dimensional axisymmetric boundaries represent three-dimensionally based wall shear rates. In the three-dimensional case, the thrombus boundary was determined through a destructive micro computed tomography (μCT) scan, limiting the study to a sample size of one. An unstructured tetrahedral mesh was applied throughout the flow domain. The same computational meshing and CFD packages were used for both two-dimensional and three-dimensional studies. Similarly mesh and solver convergence criteria were the same for the three-dimensional case. Flow into the stenosis was set at 0.1 ml/min, corresponding to a pressure drop of 45 mmHg across the thrombus and stenosis. The flow rate is lower than the 5 cases of thrombus growth in thus study, but the pressure drop is the same. The μCT case also has a more severe stenosis, which

combined with the flow rate, result in shear rates that are on the same order as shear rates in the 5 main cases. This three-dimensional study is not included in relating shear to thrombus growth, but is instead described to demonstrate the three-dimensional impact on wall shear rates. For comparison, a projection of the three dimensional thrombus was made onto a two-dimensional plane. The lumen radius based on the projection was used to analyze shear under the assumption of axisymmetry to mimic the two-dimensional images taken by light microscopy in the main 5 cases.

G. Thrombus growth rate quantification

To characterize thrombus growth relative to shear, a thrombus growth rate was calculated, which normalizes thrombus growth with deposition area and time. Thrombus growth rate, J, is calculated by differentiating thrombus growth with respect to time and dividing by the local instantaneous surface area of the growing thrombus. The growth rate is mathematically described as:

Equation 3.3

$$J = \underbrace{\frac{1}{2\pi \underbrace{\left(y'_0 - \tau\right)}_{\text{Inner Radius}} dz}}_{\text{1/(Thrombus Deposition Surface Area)}} \underbrace{\frac{d}{dt}\pi\left(y'^2_0 - \underbrace{\left(y'_0 - \tau\right)^2}_{\text{Inner Radius}}\right)dz}_{\text{Thrombus Volume}} = \dot{\tau}$$

where R'_0 is the initial lumen radius, τ is the thrombus thickness, and dz is the local axial length spanning the region of interest. Since we have discrete data, we applied the upwind differentiating scheme to obtain the discretized thrombus growth rates:

Equation 3.4

$$J_i = \underbrace{\frac{\left(y'_{0,i} - \tau_i^n\right)}{\left(y'_{0,i} - \tau_i^{n-1}\right)}}_{\approx 1 \text{ for small change in } y_i} \frac{\tau_i^n - \tau_i^{n-1}}{\Delta t}$$

The superscript *n* denotes the time step, while the subscript, *i*, denotes the axial point. The images in this study are divided into approximately 1300 axial points and Equation 3.4 is applied at each point. Additional platelet deposition rates are estimated by using a multiplication factor of $f/V_{platelet}$, to convert thrombus growth rates to an approximate platelet aggregation rate, which assumes a platelet volume fraction, f=0.8, a value previously found through histology [11].

H. Model of Occlusion

Time to occlusion can be used as a measure of thrombosis risk, as past reports have indicated that probability of thrombus occlusion drops as time to occlusion increases [15]. A simple model of thrombosis risk is considered here based on predicted thrombus occlusion times to illustrate the utility of relating thrombus growth rates to shear rates. Occlusion times were calculated based on the functional relationships established in this study between thrombus growth rates and shear rates. We used a summation of two terms to obtain the time to occlusion, an initial lag time ($t_{lag\ time}$) found as thrombus grows over the collagen (growth <10 μm) and a subsequent growth of thrombus on previously deposited thrombus (growth >10 μm). Both terms are deemed to be functions of shear rates, $\dot{\gamma}_{wall}$, which vary with stenosis severity. As such, thrombus growth rates are

integrated from the initial stenosis radius, *r*, to full occlusion, while the lag time was added to obtain the total occlusion time for a specific stenosis:

Equation 3.5

$$t_{occlusion} = \int_0^r \left[\frac{1}{J\left(\dot{\gamma}_{wall}\right)} \right] dr + t_{lag\ time}\left(\dot{\gamma}_{wall,initial}\right)$$

This model of occlusion utilizes shear rates approximated through Hagen-Poiseuille flow with a parabolic velocity profile ($\dot{\gamma}_{wall} = 4Q/\pi r^3$) instead of the full CFD solution. Such flow may be considered for gradual smoothly varying stenosis, while neglecting blood pulsatility. To determine the flow rate for a specific stenosis, we use a curve fit ($Q=Q_0(1.0+1.9\text{x}10^{-2}(st)+6.2\text{x}10^{-4}(st)^2-5.2\text{x}10^{-6}(st)^3)$) from Gould et al. [31] that is based on flow through a canine left circumflex coronary artery, while assuming a nominal average flow rate of $Q_0 \approx 2$ ml/s. The term st is the stenosis severity in terms of percent by diameter.

I. Calculations of Drag Forces and Contact Times

To calculate the length of time available for a platelet to develop bonds, we determined the contact time ($T=a/v$) during which a platelet travels its own diameter, *a*, of 2.5 microns away from the wall based on the local speed of the fluid, *v*. Once a platelet travels its own diameter, the initial binding sites on a surface will have passed by the platelet and only new binding sites would be available for platelet binding. Once a platelet binds, we determine the drag force (F=2.35*dynamic viscosity*$\dot{\gamma}_{wall}$ πa^2 [32]) on a platelet based on Stokes flow corresponding to a particle Re that is less than 1.

Results

A Coulter counter was used to quantify the average mean platelet volume ($V_{platelet}$=8.8±1.0 μm^3), the average platelet concentration (c_0=210±74 million platelets per milliliter), and average hematocrit (45±9.5%), among other variables.

A. Time Dependent Thrombus Growth

Since thrombus growth may vary with time and space along a stenosis, we evaluate growth temporally and spatially over five cases. For thrombus growth measurements, the thrombus edge was determined along the entire axis of the stenosis approximately every second. The most substantial thrombus growth was found near the stenosis throat for all five cases throughout the time series of images studied here, as illustrated in Figure 3.2A-E. Most growth was found within 1 nominal vessel diameter in either direction of the stenosis apex. Only trace amounts of thrombus were found more than one diameter downstream of the stenosis. No thrombus was found upstream to the stenosis where no collagen coating existed.

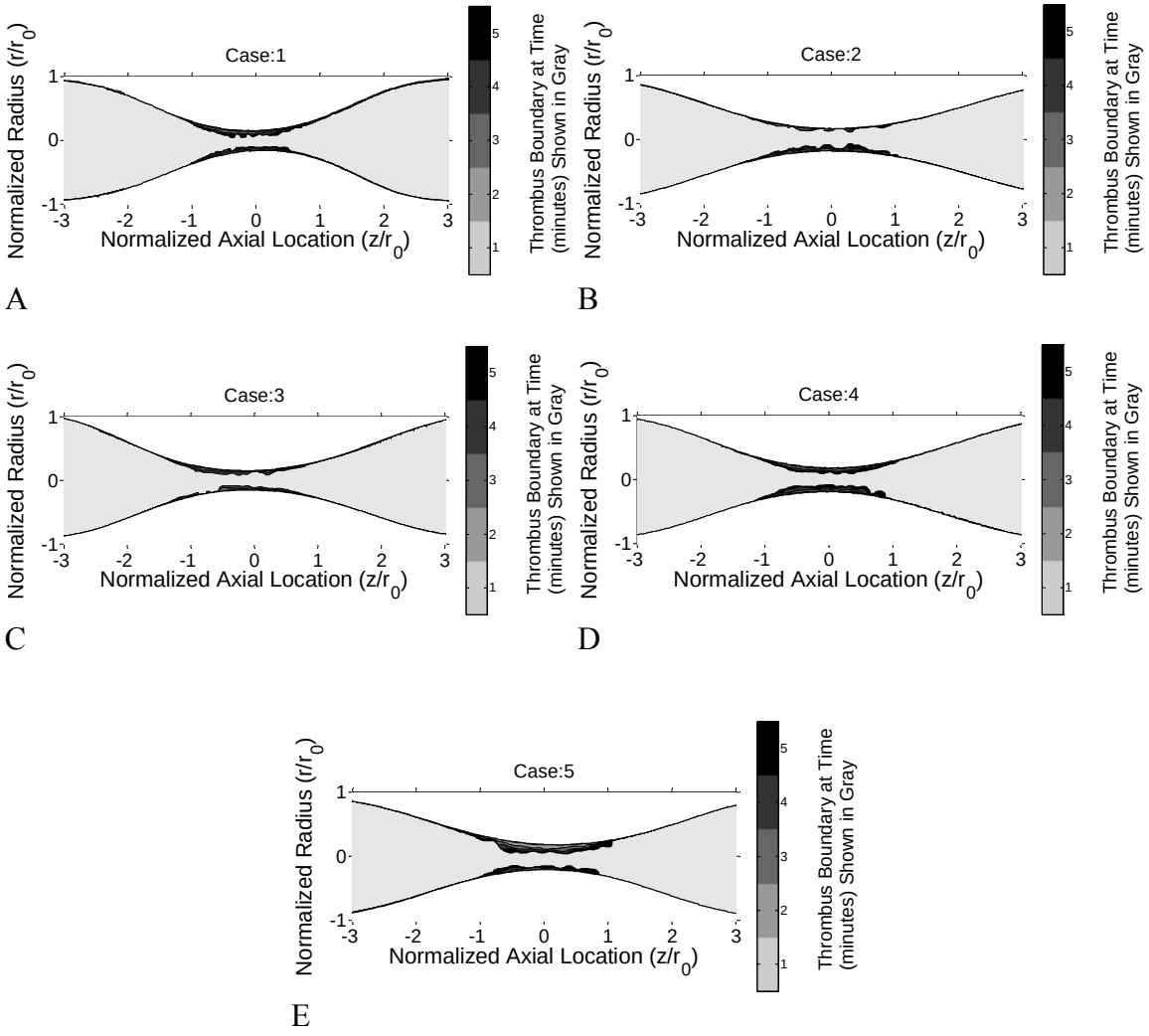


Figure 3.2: Thrombus growth determined from edge detection with a corresponding magnified view. Thrombus growth is plotted as contours illustrating spatial and temporal thrombus growth. A) Case 1 B) Case 2 C) Case 3 D) Case 4 E) Case 5

After 10 microns of growth, the average thrombus thickness ramps up to the must substantial growth at half of a nominal radius upstream to the stenosis apex, as illustrated in Figure 3.3. There is a moderate decreasing gradient of thrombus from the maximal thickness location until half of a nominal radius downstream of the apex. Additionally, on average, it took 2 minutes to reach 10 microns of thrombus growth, while it only took

another 80 seconds to reach 45 microns of growth for the region of maximum thrombus thickness, illustrating an accelerated growth after an initial lag time, which is also demonstrated Figure 3.4. The figure depicts the change in thrombus thickness with respect to time at the apex of the stenosis for all 5 cases of growth that were studied. The lines correspond to a 10 micron wide region. The trends were similar between all 5 cases.

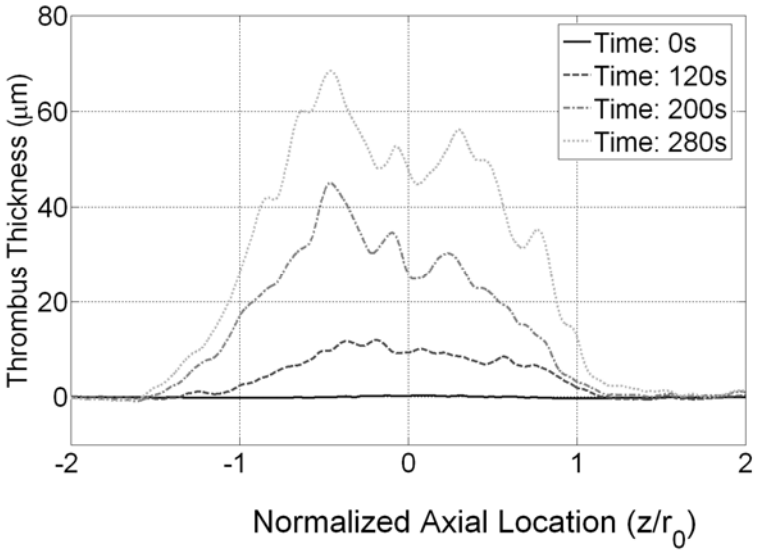


Figure 3.3: Average thrombus thickness for all cases across the stenosis for different axial location.

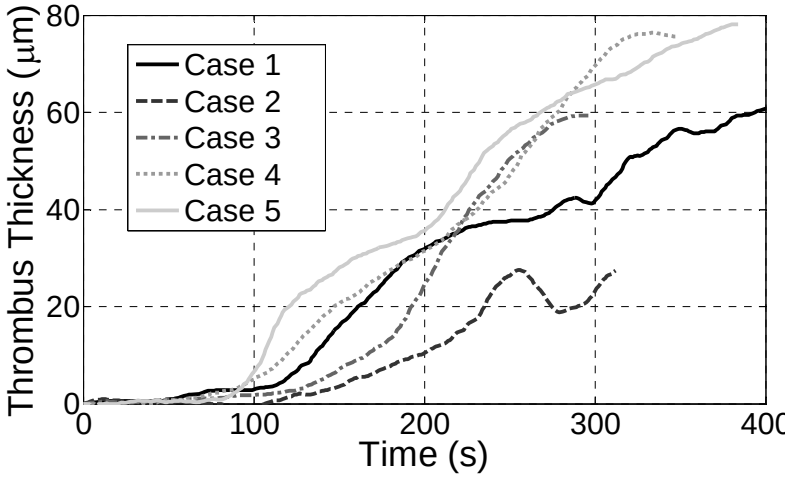


Figure 3.4: Thrombus thickness at the stenosis apex, illustrating an initial time lag of growth, with subsequent rapid thrombus growth. All cases had similar trends. The growth rate is calculated based on differentiating the curve of thrombus thickness with respect to time.

B. Wall shear rates during thrombus growth

Throughout the growth process, the hemodynamics near the surface change due to the changing lumen diameter. To determine variations in wall shear rates, thrombus edges were applied as a CFD wall boundary every 10 seconds. Spatially, initial wall shear rates average near 20 s^{-1} in non-stenosed regions and increase to 5,000 s^{-1} near the throat of the stenosis, as presented in Figure 3.5A-E. Maximum shear rates and the initiation of the experiment and termination of the experiment are given relative to the corresponding stenosis severity in Table 3.2. Shear rates increased by a factor of 20 as thrombus grew from the initial surface until a pressure drop of 50 mmHg across the stenosis, where wall shear rates reached up to 100,000 s^{-1}. Shear rates at the end of each experiment exhibited spatial undulations in regions of thrombus protrusions. Flow rates in this study remain constant and therefore decreasing lumen diameters from thrombus growth cause large increases in wall shear during the experiments. Blunted velocity profiles, reminiscent of entrance flow exist at thrombus protrusions, causing further increases in shear. Flow separation due to adverse pressure gradients only occurred in Case 4 and did so downstream to a large thrombus protrusion, where a large diverging geometrical gradient exists.

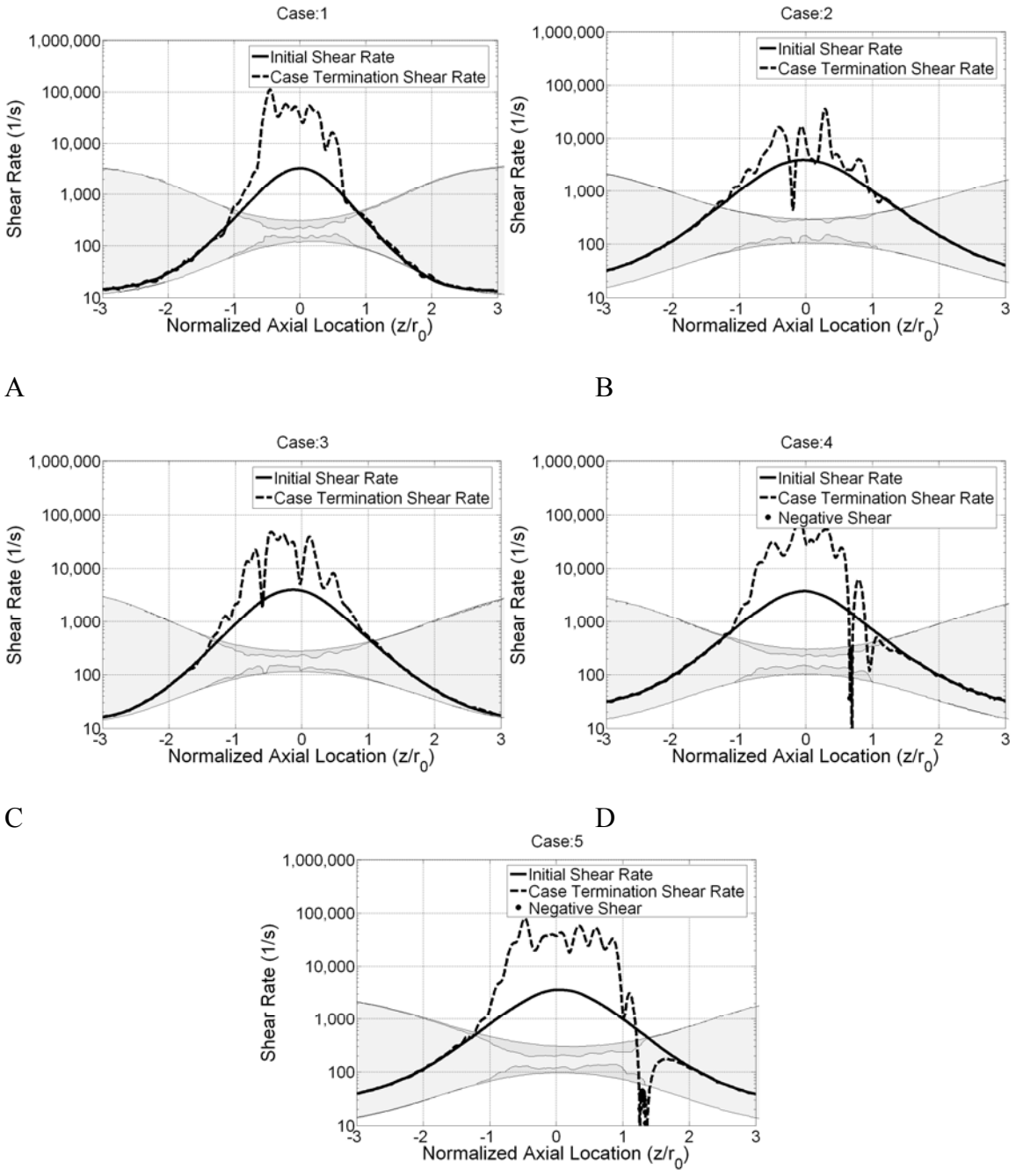


Figure 3.5: Wall shear rates along the stenosis axis for the initial stenosis geometry and the final stenosis geometry. The edges detected for the stenosis and thrombus are plotted in the background of the images as a reference. A) Case 1 B) Case 2 C) Case 3 D) Case 4 E) Case 5

Table 3.2: Experimental conditions at beginning and end of experiment

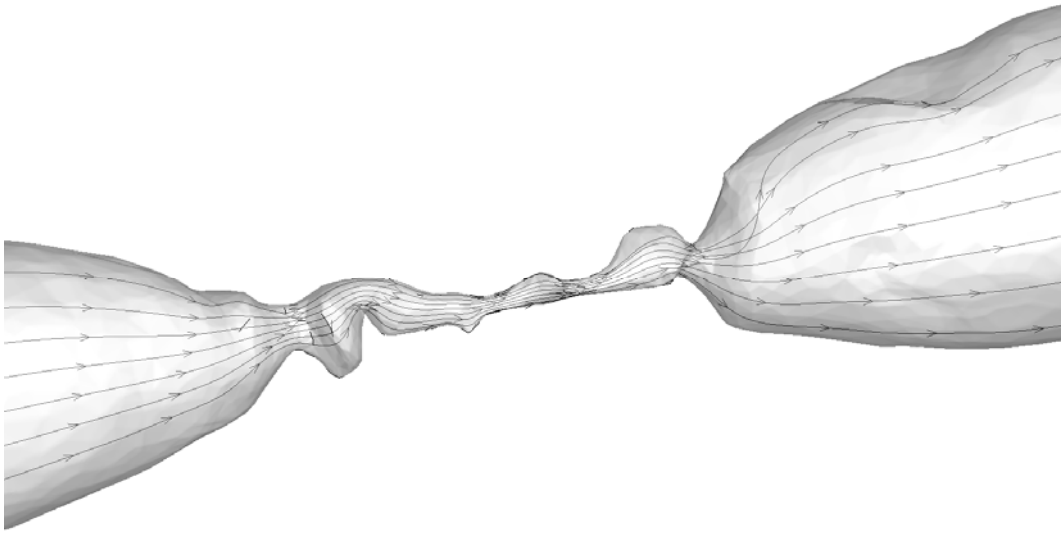
Case	$Stenosis_{initial}$ (%)	$Stenosis_{final}$ (%)	$Shear_{initial}$ (1/s)	$Shear_{final}$ (1/s)
1	84	96	3,300	110,000
2	82	91	3,900	36,000
3	85	94	4,000	48,000
4	81	93	3,800	84,000
5	80	93	3,600	79,000

C. Comparison between two-dimensional and three-dimensional shear computations

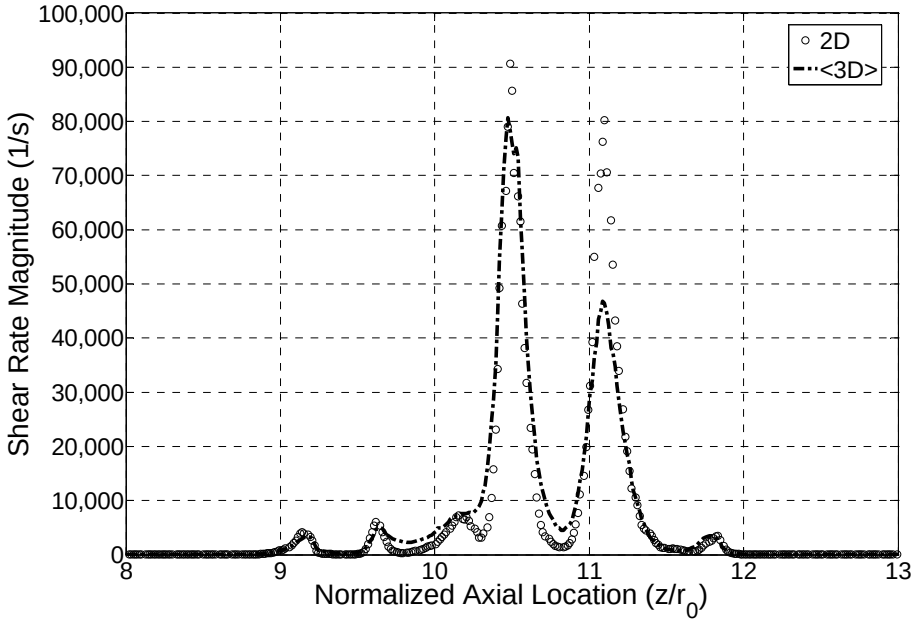
Thrombus growth and the corresponding flow field are three-dimensional. To demonstrate the efficacy of estimating three-dimensional wall shear rates from two-dimensional projections under the present flow conditions, the wall shear rates were compared for a two-dimensional and three-dimensional case. For the three-dimensional case, thrombus morphology was reconstructed from microtomography. Growth resulted in thrombus protrusions, which created a series of peaks and valleys on the surface that were not always axisymmetric, as can be seen in Figure 3.6A. Fluid streamlines are presented in Figure 3.6A to illustrate the three-dimensional flow patterns. Flow, like the geometry, does not follow axisymmetric characteristics, as illustrated by tortuous streamlines. Spatially averaged hemodynamics are studied here to determine how well two-dimensional projections represent three-dimensional flow fields for the thrombosis in this study.

To compute the flow field for the two-dimensional case, an axisymmetric geometry was created based on a projection of the three-dimensional case onto a plane. Wall shear rates were determined from boundaries defined by the projection, as

illustrated in Figure 3.6B. Plotted on the same figure, are circumferentially spatially averaged wall shear rates of the three-dimensional case. Effects from eccentricity and asymmetry are studied by comparing the two curves. The axisymmetric model captures many of the wall shear rate features of the three-dimensional model with some differences in wall shear rate magnitude. Correspondingly, the thrombus does not grow uniformly and smoothly. As such, thrombus growth in-plane of the projection may not grow to the same thrombus thickness of out-of-plane sections. On average, the projection shear rates were within 15% of the three-dimensional circumferentially averaged shear rates.



A



B

Figure 3.6: A) Streamlines through a three-dimensional reconstruction of thrombus through μCT. B) Wall shear rates relative to axial location for a two-dimensional project of the three-dimensional geometry and circumferentially averaged wall shear rates relative to axial location for a three-dimensional reconstruction of thrombus.

D. Relationship between thrombus growth rates relative to thrombus thickness, wall shear rate, lumen radius, axial location, time, and the wall shear rate gradient

Growth in the test section clearly exhibited a slow initial deposition phase, then a more rapid accumulation phase. To study possible hemodynamic effects on thrombus growth rates, we perform a regression analysis between large scale thrombus growth rates (growth >10 μm) and shear during acumulation. Subsequently, we evaluate growth during initial deposition that is less than 10 μm. Lastly we consider the correlation between growth rates and variables other than shear.

The rapid growth that occurs over previously deposited thrombus (>10 μm) is shown in Figure 3.7. The plot consists of wall shear-averaged data among all 5 cases of thrombus growth studied here. Spatial and temporal data was considered independent for the averaging. A linear correlation ($r=0.56$, $p<0.0001$) of growth rates relative to shear exists until 6,000 s^{-1} for all points. After a log-log transformation, a strong correlation was found ($r=0.85$, $p<0.0001$). The maximum average growth rate reached 20 $\mu m^3/\mu m^2$-min at 6,000 s^{-1}, although some individual points exhibited rates as high as 30 $\mu m^3/\mu m^2$-min. Higher shear above 6,000 s^{-1} results in a decreasing trend of thrombus growth rates relative to shear rate with a weak negative correlation ($r = -0.26$, $p<0.0001$). Nonlinear transformations resulted in correlation coefficients that were negligibly better. Overall, thrombus growth rates for pathophysiologic arterial shear rates (>1000 s^{-1}) resulted in high thrombus growth rates that were 2-4 times greater than for physiological arterial shear rates.

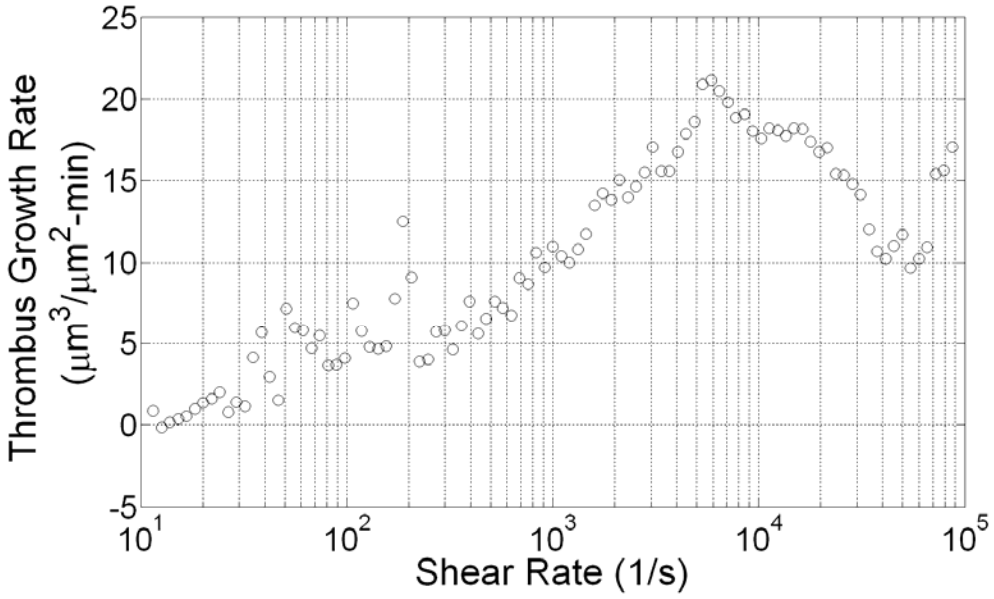


Figure 3.7: Mean thrombus growth rates with corresponding shear rates. The data spans 5 experiments with approximately 5,000 samples per experiment. Data is assumed to be spatially and temporally independent.

To further evaluate the relationship between thrombus growth rates and wall shear rates, we performed regression analyses and evaluated the distribution of growth rates for a range of shear rates. The relatively good correlation for a log-log transformation corresponds to a power-law regression between thrombus growth rates and wall shear rates. For shear rates less than 6,000 s^{-1}, a power curve

($J_{(<6000\ s^{-1})} = 0.41(\dot{\gamma})^{.45}$ $\mu m^3/\mu m^2$-min) was found. Since the correlation for a nonlinear transformation was only marginally better than a linear correlation for higher shear rates, a nonlinear regression was not performed for higher shear. The linear regression for shear rates exceeding 6,000 s^{-1} was found to be

$J_{(>6000\ s^{-1})} = 20 - (2.1x10^{-4})\dot{\gamma}$ $\mu m^3/\mu m^2$-min. The power law regression for low shear rates and linear regression for high shear rates are illustrated in Figure 3.8A.

From Figure 3.8A, it can be seen that the residuals to the regression line are larger for low shear (30-200 s^{-1}) and extremely high shear (>80,000 s^{-1}). Figure 3.8B illustrates that these regions also have the lowest number of samples. As a statistical check of the distribution of the data points around the mean, we plotted a histogram of growth rates for a shear rate range of 4,000-8,000 s^{-1},where the number of observations is high. The histogram is contained within a Gaussian shape (r^2=0.99), as illustrated in Figure 3.8C. The Gaussian fit provides evidence that the mean of our sample is representative of a population mean.

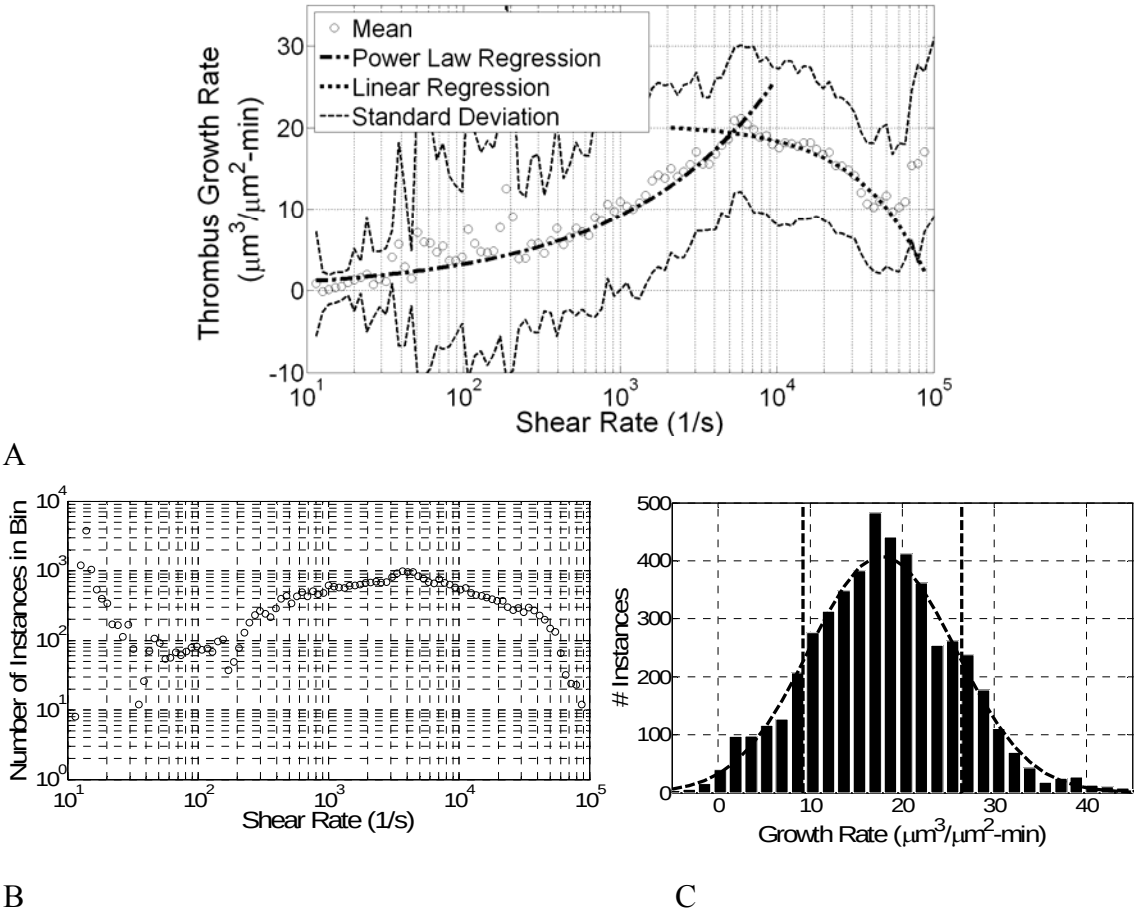


Figure 3.8: A) Plot of the power law fit for data that at shear rates lower than 6,000 s^{-1}, with a linear fit taken for data greater than 6,000 s^{-1}. Included in the plot are mean thrombus growth rates and the standard deviations. B) The number of instances that a particular shear rate was found for thrombus thickness that was greater than 10 microns. A higher number of instances would correspond to an increased likelihood that thrombus growth rate predictions represent the population. C) The distribution of growth rates for a range of shear to illustrate the Gaussian distribution that can be seen for the data in shear bins.

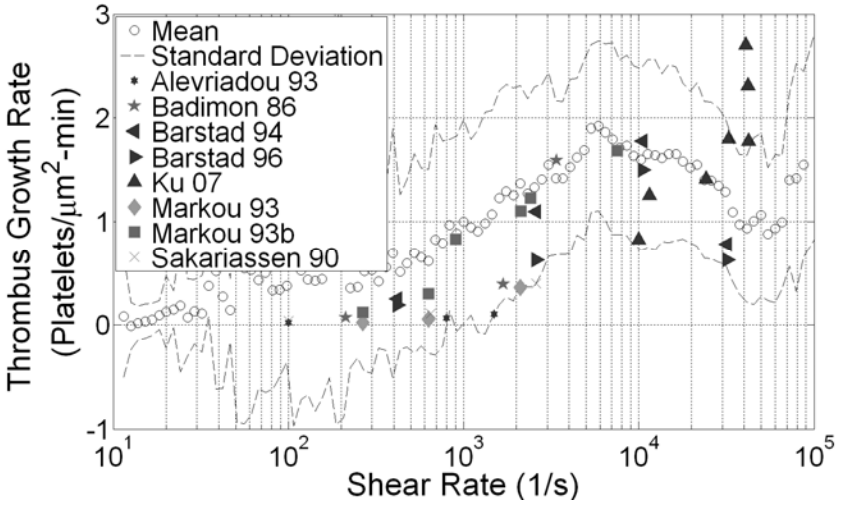
To relate our work to the field, we developed a conversion of thrombus growth rates, J, to platelet accumulation rates, $Jf/V_{platelet}$, as illustrated in Figure 3.9A. Values plotted from the literature in the figure include many different species and experimental techniques, as shown in Table 3.3. Platelet accumulation in our study generally corresponded to the upper limit of values reported in the literature. Most previous reports fall within a standard deviation of our mean platelet accumulation values.

Table 3.3: Experimental conditions of previous studies found in the literature

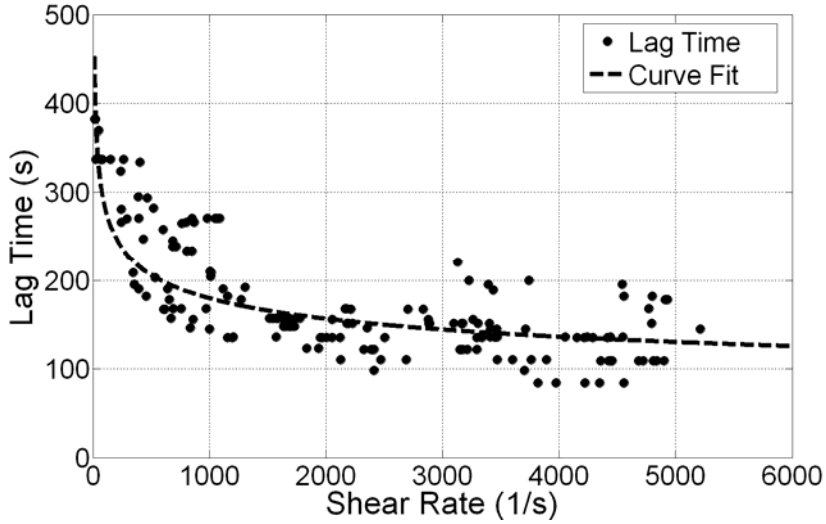
Ref	Blood	Method	Initial Surface	Measure
Alevriadou [22]	Human	Parallel plate, *in vitro*	Collagen type I	Fluorescent intensity
Badimon [18]	Porcine	Tube, *in vitro*	Deendothelialized & Collagen type I	Radiolabel
Barstad '94 [19]	Human	Parallel plate, *ex vivo*	Collagen type III	Light intensity
Barstad '96 [20]	Human	Parallel plate, *ex vivo*	Collagen type III	Light intensity
Ku [2]	Porcine	Tube, *in vitro*	Collagen type I	Effective orifice area
Markou '93 [17]	Baboon	Tube, *ex vivo*	Collagen	Radiolabel
Markou '93b [10]	Baboon	Tube, *ex vivo*	Collagen	Radiolabel
Sakariassen [23]	Human	Parallel plate, *ex vivo*	Collagen type III	Light intensity

Before platelet deposition reached 10 μm in thickness, the thrombus exhibited a lag time where very little growth occurred, as seen in Figure 3.4. Thrombosis during the lag time consists of growth over the initial collagen coated surface, whereas subsequent thrombi form on already deposited thrombus. The lag time decreased as shear rates increased, as can be seen in Figure 3.9B. We saw a negative correlation of lag time relative to shear rates ($r=-0.72$, $p<0.0001$). The best correlation was found for a log-log

transformation (r=-0.85, p<0.0001). A power-law regression fit the data well, as can be seen in Figure 3.9B ($t_{\text{lag time}} = 718(\dot{\gamma})^{-0.2}$ seconds). The lag time is very long in the arterial physiological range (<400 s^{-1}) suggesting little thrombus generation, with lag times of less than 4 minutes for pathologic shear rates.



A



B

Figure 3.9: A) Platelet accumulation rates predicted from thrombus growth rates. The standard deviation is also illustrated to demonstrate that the previously reported values of platelet deposition rates typically fall within the bounds of our standard deviation. B) Lag time of thrombus growth until a thrombus thickness of 10 microns, with a corresponding power curve fit. The shaded region illustrates the physiologic range of shear for the coronary artery.

In order to evaluate whether other variables have a major impact on thrombus growth rates, we determined correlation coefficients of thrombus growth rates (growth >10 μm) relative to: thrombus thickness ($r=0.47$, $p<0.0001$), lumen diameter ($r=-0.55$, $p<0.0001$), axial location ($r=-0.08$, $p<0.0001$), time ($r=-0.30$, $p<0.0001$), and no significant correlation was seen for wall shear rate gradients on the macro-scale. All of these variables resulted in lower correlation coefficients than what was found for thrombus growth rates relative to shear rates, except for the lumen diameter, which is functionally associated with wall shear rates.

E. Quantification of occlusion time for an idealized gradual stenosis in a coronary vessel based on average conditions

To illustrate the utility of relating thrombus growth to shear rates, a simple model of thrombosis risk is considered here based on predicted thrombus computational occlusion times. If we only consider radial changes at the stenosis apex, then the time to computational occlusion is represented in Figure 3.10. Assuming that thrombus growth rates can be predicted from the regression analysis presented in the current study, then thrombus at shear rates less than 6,000 s^{-1} follow a power curve ($J_{(<6000\ s^{-1})}$) and higher shear rates follow a linear curve ($J_{(>6000\ s^{-1})}$) until 50,000 s^{-1}. At extremely high shear rates, the thrombus in the model is assumed to grow at the same rate as 50,000 s^{-1}, because data is too sparce, as was seen in Figure 3.8B. The time to compuataional occlusion model also includes the lag time $t_{lag\ time}$ (T). The time to computational occlusion decreases very rapidly for stenoses of 88% by diameter and higher. Compuataional occlusion times are predicted to be on the order of minutes for very

severe stenoses, but would be hours for stenoses that are less than 75% by diameter. The model of occlusion time does not account for thrombolysis.

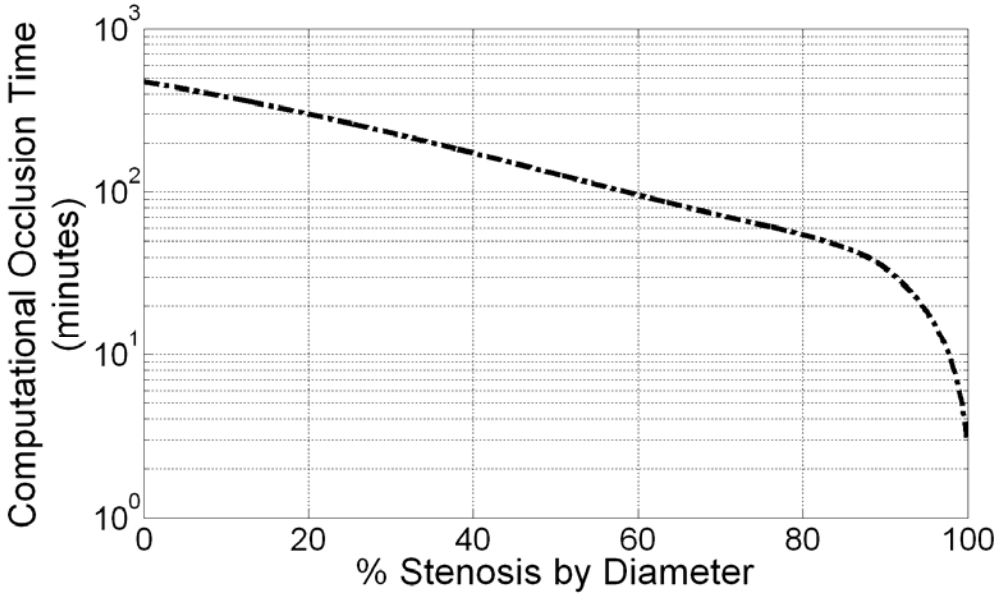
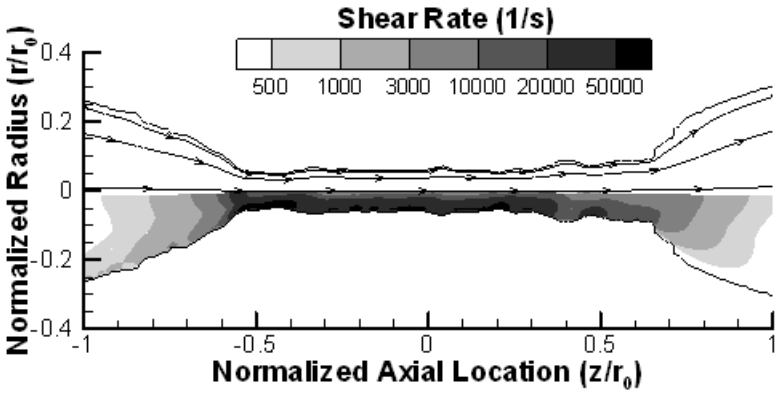


Figure 3.10: Time to computational occlusion relative to stenosis severity. The stenosis severity is related back to shear rates to predict thrombus growth rates and lag times that combine for computational thrombus occlusion time predictions.

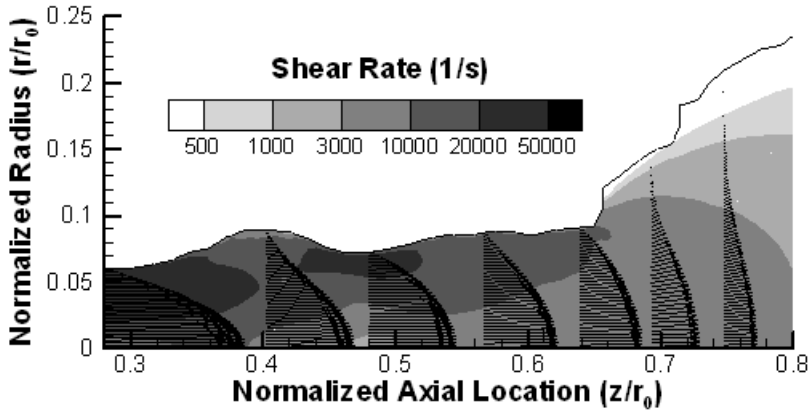
F. Platelet Forces and Contact Times during thrombus growth

For thrombus to grow, platelets must attach under the hemodynamics studied here. Blood traveling close to the wall convects circulating platelets that are about to stick to the surface. As shown in Figure 3.11A, streamlines approach high shear at peaks in the undulating thrombus surface, but diverge from valley regions of the surface. In these regions of very high shear (100,000 s^{-1}), platelets traveling in the flow 2.5 microns away from the wall (a platelet's average diameter) would have a speed of 25 cm/s. Thus, platelets would traverse their own diameter very quickly, in approximately 10 μs. Since thrombus was found to locally grow in regions of up to 100,000 s^{-1} (15,000 dynes/cm^2),

platelets must be forming bonds at these speeds. The stenotic region is where the thrombus forms most prevalently consistent with location where platelets move at high speeds as shown in the velocity profiles of Figure 3.11B.



A



B

Figure 3.11: A) Streamlines and shear rate contours for flow through thrombus near the apex of Case 1 after thrombus had formed for 7 minutes. The shear along the surface is not uniformly distributed. B) Velocity vectors and shear contours, illustrating the complexity of the flow near the growing thrombus surface.

Platelets attaching to the wall at high shear rates must endure high drag forces. For a shear rate of 100,000 s^{-1}, a drag force of 17,500 pN would be exerted upon an attached platelet, assuming a viscosity of 3.8 cP and a platelet radius of 1.25 μm

Discussion

Average wall shear rates in the arterial system have been reported as 400 s^{-1}, a value consistent among species and arteries [33]. Deviations above this value result in thrombus growth rates that increase with wall shear, until a wall shear rate of 6,000 s^{-1} under the experimental conditions described in the present article. Further increases in shear results in a decreasing growth rate trend until at least 100,000 s^{-1}, yet thrombus growth continued with a higher growth rate that is 2-4 times greater than what was found at 400-1,000 s^{-1}. Therefore, medical device designs, such as left ventricular assist devices or heart valves may form thrombus if shear rates reach 6,000 s^{-1}. Thrombus would also be less likely to grow quickly in the shoulder of an atheroma where plaque-cap rupture is commonly found to occur, as compared to rupture or erosion near the apex of a lesion

Our results that consider temporally and spatially varying shear rates are consistent with results from previous studies that evaluated thrombus growth at spatially constant initial wall shear rates. Notably, results from *ex vivo* baboon shunts match particularly well with our mean growth rates for a given shear (<7,500 s^{-1}) [10], with a growth rate of 0.83, 1.23, and 1.69 platelets/μm^2-min relative to a shear rate of 900, 2,400, and 7,500 s^{-1} respectively. Comparably, the present study has respective growth rates of 0.88, 1.27, and 1.71 platelets/μm^2-min. Another set of studies using human blood in a parallel plate system found an increasing trend of thrombus growth rates until 10,000 s^{-1} and then found that growth rates drop at 32,000 s^{-1}. [19-20] , with a growth rate of 1.09, 1.78, and 0.78 platelets/μm^2-min with respective shear rates of 2,600, 10,500, and 32,000 s^{-1}. We found growth rates of 1.35, 1.65, and 1.20 platelets/μm^2-min, respectively, that exhibited a similar trend of increasing, and then decreasing growth rates

relative to shear. Contrary to previous methods, the present study measures thrombus growth rates and the corresponding local shear rate that are updated with the changing thrombus thickness. The current methods allow us to evaluate growth rates for a large sample number (25,000) for a large range of shear (10-100,000 s^{-1}), whereas previous studies have much lower sample sizes where separate experiments were required for each evaluated shear rate.

The transition from an increasing to a decreasing trend of thrombus growth rates relative to shear may be explained by a transition from one rate limiting mechanism to another. Increasing trends of thrombus growth rates relative to shear could be explained by either increasing transport [34] from enhanced red blood cell motion or increasing kinetic rates that can occur with catch bonds [35]. The axial decrease of thrombus growth in the flow direction that was seen in Figure 3.3 is a characteristic of transport limited processes causing depletion of platelet concentration in the boundary layer when moving axially downstream of flow, as noted by Turitto et al. [24]. Therefore, transport may play a significant role in thrombus growth.

Decreasing thrombus growth rates relative to shear can similarly be explained by decreases in transport or kinetic rates. Although we found no studies that consider transport for shear rates exceeding 5,000 s^{-1}, particle collisions per unit time are reported to theoretically scale with shear rates without bound [36]. Therefore kinetics may be limiting thrombus growth rates with high shear (>6,000 s-1). The transition from a catch bond to a slip bond can cause a transition from increasing to decreasing bond lifetimes [35] and thus could result in a similar transition in platelet deposition rates relative to shear. Increases in the number of bonds or the strength of bonds may be required to

sustain increased platelet drag forces caused by increases in wall shear. Binding of platelet glycoprotein (GP) Ibα to the vWF A1 domain, is reported as a necessary condition for platelet binding at high shear [37]. The single GPIbα-A1 tether force is on the order of 70 pN [35, 38] for the longest bond lifetime as opposed to 17,500 pN that may exist for platelets with multiple GPIb ligands or very short bond lifetimes at the highest shear rates considered in the present study. Another issue is that the time available for a bond to form between a platelet and another surface will decrease with increasing shear due to increases in near-wall flow speed. Platelets traveling at the fluid speed traverse their own length within 10 μs for the highest shear considered in this study.

The mechanisms associated with the lag time seen in this study may represent platelet adherence to the collagen surface. The lag time trends relative to shear correspond with reported activation time trends [39]. Hellums et al. report a decreasing time to activation for increases in shear rate. Their data analysis indicates an activation time near 100 s for a shear rate of 6,000 s^{-1}, which is near the lag time we see for the same shear rate. Activation may promote adhesive capabilities of platelets through agonist release and activation of GPIIb/IIIa, a glycoprotein believed to allow more stable platelet adhesion. Platelets also contain a vWF concentration that is a factor of 50 higher than vWF concentrations in plasma [40], even though only a fraction is believed to be released from α-granules upon platelet activation. To this regard, platelet endogenous vWF has been reported to have the capability of playing a major role in shear-dependent platelet accumulation [41]. Activation of thrombus platelets is one possible explanation for the increased thrombus growth rates after an initial lag time during blood perfusion.

The resultant relationship between thrombus growth rates and shear can be extended to develop predictive models of thrombus growth, as was done with the simplified occlusion model presented here. Computational occlusion time can be used a measure for assessing a stenosis risk for thrombus occlusion. As occlusion time increases, there is more time for thrombus lysis and intervention. While the present model is very limited in nature, it does show that increasing stenosis severity corresponds with decreasing time to occlusion. More complex computational models can be implemented that take into account complex flows based on patient-specific information, while relating corresponding shear rates to thrombus growth through the relationship found in the present study. The more complex computational model would account for thrombotic factors that can vary among patients, such as platelet concentration, or activation agonist activity [42]. Such computational models could also account for thrombolysis.

This study has several limitations. All test sections in this study are approximately the same, with an increasing shear rate with respect to time as thrombus grows. Therefore, we don't evaluate thrombus growth rates for shear rates that decrease with time. We also don't vary flow rates or geometry. Our results are, however, consistent with the literature that has investigated different flow rates and geometry. Shear rates in this study are quantified under two-dimensional axisymmetric assumptions, even though thrombus may be irregular, with shear that differs on average by 15% from three-dimensional computations. Furthermore, shear predictions do not consider flow through the porous thrombus structure. Calculations did not take into account the settling of particles in the syringe pump over time. Many factors increased the variability in this study. Platelet concentrations, hematocrit, and platelet volume all had high variability

and were not standardized across cases in this study. Previous studies have shown, for example, that platelet concentration can impact thrombus growth [43]. We also did not account for the age of blood after collection and the concentration of binding factors. Contact with silicone and Pyrex® and handling of the blood may affect platelet activation states. Additionally, results may vary if we use human blood instead of porcine blood. Various binding proteins associated with thrombosis, such as vWF, may also be released in high concentrations due to the stress placed on the pigs before we obtain the blood used in the present study.

Conclusion

Thrombus growth rates were found to correlate strongly with wall shear rates when comparing wall shear rates predicted from CFD to growth quantified by detecting the edge of growing thrombi over a stenosis. We also show that thrombus continues to grow for very high shear rates of up to 100,000 s^{-1} although with some diminishment at extremely high shears. Nonetheless, thrombus accumulates at rates 2-4 times higher than normal for all shears rates greater than 400 s-1. The quantitative relationship can be used to predict thrombus location and time to occlusion for flowing blood.

References

1. Wootton, D.M., et al., *A mechanistic model of acute platelet accumulation in thrombogenic stenoses.* Annals of biomedical engineering., 2001. **29**(4): p. 321-329.
2. Ku, D.N. and C.J. Flannery, *Development of a flow-through system to create occluding thrombus.* Biorheology, 2007. **44**(4): p. 273-284.
3. Yutani, C., et al., *Coronary atherosclerosis and interventions: pathological sequences and restenosis.* Pathology International, 1999. **49**(4): p. 273-290.
4. Davies, M.J. and A. Thomas, *Thrombosis and acute coronary-artery lesions in sudden cardiac ischemic death.* N Engl J Med, 1984. **310**(18): p. 1137-40.
5. Viles-Gonzalez, J.F., V. Fuster, and J.J. Badimon, *Atherothrombosis: a widespread disease with unpredictable and life-threatening consequences.* European heart journal., 2004. **25**(14): p. 1197-1207.
6. Chandler, A., et al., *Coronary thrombosis in myocardial infarction:: Report of a workshop on the role of coronary thrombosis in the pathogenesis of acute myocardial infarction.* The American Journal Of Cardiology, 1974. **34**(7): p. 823-833.
7. Folts, J.D., E.B. Crowell, Jr., and G.G. Rowe, *Platelet aggregation in partially obstructed vessels and its elimination with aspirin.* Circulation., 1976. **54**(3): p. 365-370.
8. Badimon, L. and J.J. Badimon, *Mechanisms of arterial thrombosis in nonparallel streamlines: platelet thrombi grow on the apex of stenotic severely injured vessel wall. Experimental study in the pig model.* The Journal Of Clinical Investigation, 1989. **84**(4): p. 1134-1144.
9. Lassila, R., et al., *Dynamic monitoring of platelet deposition on severely damaged vessel wall in flowing blood. Effects of different stenoses on thrombus growth.* Arteriosclerosis : an official journal of the American Heart Association, Inc., 1990. **10**(2): p. 306-315.
10. Markou, C.P., et al. *Role of high wall shear rate on thrombus formation in stenoses*. in *Advances in Bioengineering*. 1993. New Orleans: ASME.
11. Flannery, C.J., *Thrombus formation under high shear in arterial stenotic flow.*, in *Mechanical Engineering*. 2005, Georgia Tech: Atlanta.
12. Bark, D. and D. Ku, *Wall shear over high degree stenoses pertinent to atherothrombosis.* Journal of Biomechanics, 2010.
13. Siegel, J.M., et al., *A scaling law for wall shear rate through an arterial stenosis.* Journal of Biomechanical Engineering, 1994. **116**(4): p. 446-451.
14. Strony, J., et al., *Analysis of shear stress and hemodynamic factors in a model of coronary artery stenosis and thrombosis.* American journal of physiology. Heart and circulatory physiology, 1993. **265**(5): p. H1787-1796.
15. Davies, M.J. and A.C. Thomas, *Plaque fissuring--the cause of acute myocardial infarction, sudden ischaemic death, and crescendo angina.* British Heart Journal, 1985. **53**(4): p. 363-373.

16. Cadroy, Y., T.A. Horbett, and S.R. Hanson, *Discrimination between platelet-mediated and coagulation-mediated mechanisms in a model of complex thrombus formation in vivo.* The Journal of laboratory and clinical medicine., 1989. **113**(4): p. 436-448.
17. Markou, C.P., Lindahl, A.K., Siegel, J.M., Ku, D.N., Hanson, S.R., *Effects of blocking the platelet GPIb interaction with von Willebrand Factor under a range of shearing forces.* Annals Of Biomedical Engineering, 1993. **21**: p. 220.
18. Badimon, L., et al., *Influence of arterial damage and wall shear rate on platelet deposition. Ex vivo study in a swine model.* Arteriosclerosis (Dallas, Tex.), 1986. **6**(3): p. 312-320.
19. Barstad, R.M., et al., *A perfusion chamber developed to investigate thrombus formation and shear profiles in flowing native human blood at the apex of well-defined stenoses.* Arteriosclerosis And Thrombosis: A Journal Of Vascular Biology / American Heart Association, 1994. **14**(12): p. 1984-1991.
20. Barstad, R.M., P. Kierulf, and K.S. Sakariassen, *Collagen induced thrombus formation at the apex of eccentric stenoses--a time course study with non-anticoagulated human blood.* Thrombosis And Haemostasis, 1996. **75**(4): p. 685-692.
21. Hubbell, J.A. and L.V. McIntire, *Platelet active concentration profiles near growing thrombi. A mathematical consideration.* Biophysical journal., 1986. **50**(5): p. 937-945.
22. Alevriadou, B.R., et al., *Real-time analysis of shear-dependent thrombus formation and its blockade by inhibitors of von Willebrand factor binding to platelets.* Blood., 1993. **81**(5): p. 1263-1276.
23. Sakariassen, K.S., et al., *Collagen type III induced ex vivo thrombogenesis in humans. Role of platelets and leukocytes in deposition of fibrin.* Arteriosclerosis, Thrombosis, And Vascular Biology, 1990. **10**(2): p. 276-284.
24. Turitto, V.T. and H.R. Baumgartner, *Platelet deposition on subendothelium exposed to flowing blood: mathematical analysis of physical parameters.* Transactions - American Society for Artificial Internal Organs., 1975. **21**: p. 593-601.
25. Savage, B., E. Saldívar, and Z.M. Ruggeri, *Initiation of platelet adhesion by arrest onto fibrinogen or translocation on von Willebrand factor.* Cell., 1996. **84**(2): p. 289-297.
26. Young, D.F., *Fluid mechanics of arterial stenoses.* Transactions of the ASME. Journal of Biomechanical Engineering, 1979. **101**(3): p. 157-75.
27. Bark Jr, D.L. and D.N. Ku, *Wall shear over high degree stenosis pertinent to atherothrombosis.* Journal of Biomechanics, **Accepted**.
28. Ma, P., X. Li, and D. Ku, *Heat and mass transfer in a separated flow region for high Prandtl and Schmidt numbers under pulsatile conditions.* International Journal of Heat and Mass Transfer, 1994. **37**(17): p. 2723-2736.
29. Zhao, R., M.V. Kameneva, and J.F. Antaki, *Investigation of platelet margination phenomena at elevated shear stress.* Biorheology, 2007. **44**(3): p. 161-177.
30. Kolesnikova, I., et al., *Determination of volume, shape and refractive index of individual blood platelets.* Journal of Quantitative Spectroscopy and Radiative Transfer, 2006. **102**(1): p. 37-45.

31. Gould, K.L., K. Lipscomb, and G.W. Hamilton, *Physiologic basis for assessing critical coronary stenosis. Instantaneous flow response and regional distribution during coronary hyperemia as measures of coronary flow reserve.* The American Journal Of Cardiology, 1974. **33**(1): p. 87-94.
32. Goldman, A.J., R.G. Cox, and H. Brenner, *Slow viscous motion of a sphere parallel to a plane wall--II Couette flow.* Chemical Engineering Science, 1967. **22**(4): p. 653-660.
33. Ku, D., *Blood flow in arteries.* Annual Review of Fluid Mechanics, 1997. **29**(1): p. 399-434.
34. Zydney, A.L. and C.K. Colton, *Augmented solute transport in the shear flow of a concentrated suspension.* PhysicoChemical Hydrodynamics, 1988. **10**(1): p. 77-96.
35. Yago, T., et al., *Platelet glycoprotein Ibalpha forms catch bonds with human WT vWF but not with type 2B von Willebrand disease vWF.* The Journal Of Clinical Investigation, 2008. **118**(9): p. 3195-3207.
36. Goldsmith, H. and S. Mason, *The flow of suspensions through tubes. III. Collisions of small uniform spheres.* Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 1964. **282**(1391): p. 569-591.
37. Ruggeri, Z.M., et al., *Activation-independent platelet adhesion and aggregation under elevated shear stress.* Blood, 2006. **108**(6): p. 1903-1910.
38. Doggett, T.A., et al., *Selectin-Like Kinetics and Biomechanics Promote Rapid Platelet Adhesion in Flow: The GPIb[alpha]-vWF Tether Bond.* Biophysical Journal, 2002. **83**(1): p. 194-205.
39. Hellums, J.D., *1993 Whitaker Lecture: biorheology in thrombosis research.* Annals Of Biomedical Engineering, 1994. **22**(5): p. 445-455.
40. George, J., *Platelet immunoglobulin G: its significance for the evaluation of thrombocytopenia and for understanding the origin of alpha-granule proteins.* Blood, 1990. **76**(5): p. 859-870.
41. Moake, J., et al., *Shear-induced platelet aggregation can be mediated by vWF released from platelets, as well as by exogenous large or unusually large vWF multimers, requires adenosine diphosphate, and is resistant to aspirin.* Blood, 1988. **71**(5): p. 1366-1374.
42. Sorensen, E.N., et al., *Computational Simulation of Platelet Deposition and Activation: I. Model Development and Properties.* Annals of Biomedical Engineering, 1999. **27**(4-12): p. 436-448.
43. Cadroy, Y. and S.R. Hanson, *Effects of red blood cell concentration on hemostasis and thrombus formation in a primate model.* Blood., 1990. **75**(11): p. 2185-2193.

CHAPTER 4. SPECIFIC AIM 3: RATE-LIMITNG MECHANISMS IN THROMBOSIS: A STUDY OF TRANSPORT AND BINDING AT PATHOPHYSIOLOGICAL SHEAR

Introduction

Various studies have shown that atherothrombosis can be found near an atheroma in 90% of the acute myocardial infarction cases [1-4]. As such, the understanding of processes controlling thrombosis is germane to developing methods of prediction and treatment of at-risk lesions. In addition, thrombosis is a major concern for cardiovascular medical devices such as heart valves and left ventricular assist devices. Predictions of thrombus growth risk based on hemodynamics would be useful towards the design and development of these devices.

Since atherothrombosis is found to be mostly composed of platelets [5], the mass transport of platelets to the thrombotic site plays an integral role in the thrombus growth and therefore is a sufficient subject to develop methods of predicting thrombus growth risk. Furthermore, the transport of platelets is believed to be most strongly influenced by the motion of red blood cells (RBCs). Specifically, RBCs enhance the motion of platelets across streamlines relative to motion resulting from thermal diffusivity [6-7]. The enhanced transport, although a convective phenomena, has been coined enhanced platelet diffusivity and can be attributed to the rotation and collisions of RBCs [8-9]. The number of particle collisions for a given amount of time has been reported to scale with shear rate and the local concentration of particles [10-13].

Additionally, there is a non-uniform high concentration of platelets in the plasma skimming layer near the wall in cylindrical vessels that has been attributed to RBCs and is known as margination [14-16]. Platelets near the wall in a cylindrical vessel may increase by approximately 5 to 17 fold relative to the average platelet concentration [14-15]. Therefore, more platelets are available for attachment at the thrombus surface compared to a uniform platelet concentration.

Platelet and RBC transport is also studied in computational models. Such models can be used for predictive purposes and for determining what variables should be experimentally studied. Lattice Boltzmann [17-18] and immersed finite element methods [19] have been used to model RBC physics on a small scale due to computational expense. On larger scales, multiphase methods have been used to study RBC motion [20] and discrete phase methods have been used to analyze platelet motion by assuming specific lift and drag forces on a particles [21-22]. The complexities of RBC motion and influence over platelets is still not well understood and therefore the computationally intensive models, although based on first principles, have yet to be fully validated for flow over large regions such as a atheroma.

Alternatively, the convection-diffusion equation can be used to model concentrated suspensions through the use of a potential function in the species flux constitutive equation. Turitto et al. implemented the convective-diffusion equation to analytically model the deposition of platelets through the use of an enhanced platelet diffusivity term and found platelet deposition at shear rates below 1,000 s^{-1} to be transport rate-limited [23]. The model was extended computationally to consider flow over complex geometry [24-26]. A subsequent model considered a small dynamic

growth of thrombus by artificially enlarging the viscosity of fluid in regions of thrombus [27]. These models accounted for margination by increasing the average concentration of platelets entering the flow, but a more recent study has investigated the influence of RBCs on the platelet concentration profile by considering a RBC dependent field potential [28].

In this study, we investigate the transport of platelets for stenotic flow by developing a computational model that utilizes the convection-diffusion equation and combines components of many of the previous models. The present model includes the ability to model dynamic thrombus growth with updated hemodynamic variables until the vessel is nearly occluded. The new model is used to investigate rate limiting processes in thrombus growth for shear rates ranging from 100 to 100,000 s^{-1}. These results are compared with an experimental study of thrombus growth over collagen under the same hemodynamic condition. We extend the model to predict occlusion times for varying degree stenoses to create a thrombus risk assessment tool.

Methods

Model Theory

The transport of platelets in blood flow consists of convection due to the fluid motion, diffusion of platelets across streamlines, and platelet kinetic reactions at a surface. These transport mechanisms can be determined through an incompressible mass balance of the species using the convection-diffusion equation, defined as:

Equation 4.1

$$\frac{\partial C_j}{\partial t} = -\nabla \bullet \vec{J}_j + S_j$$

where t is time, C_j is the species concentration, S_j is a source term, and $\vec{J}_j$ is the flux of platelets through an infinitesimal domain. Here, we determine the flux of RBCs and platelets based on convection and diffusion:

Equation 4.2

$$\vec{J}_j = -\vec{v}C_j - D_j \nabla \psi_j$$

where D_j is the diffusivity of the species and $\vec{v}$ is the velocity vector. The velocity is determined by solving for the incompressible conservation of mass and momentum in the Navier-Stokes equations, assuming that particles do not lag or lead flow.

The gradient in the field potential of the diffusion constitutive equation was determined based on an equation developed by Hund and Antaki [28]. A field potential acts a driving force for particles. For example, Fick's first law uses the concentration of particles as the field potential. For platelets and RBCs, the potential can be thought of as a variable that describes the number of particle collisions. The method of Hund and Antaki excludes platelets from regions of high hematocrit:

Equation 4.3

$$\nabla \psi_j = \nabla \left(C_j + C_k C_j \right)$$

where C_k is the concentration of another particle, or the concentration of RBCs in the case of blood flow. In the absence of red blood cells, Equation 4.3, assuming the platelet concentration is C_j, simplifies to the platelet gradient as the only contributory term to the field potential. Various studies have used this simplification to model platelet transport [24-27]. Inclusion of the RBCs enhances the flux of platelets based on the theory that there will be a higher number of interactions for a platelet as RBCs crowd the region around a platelet. The field potential for RBC species transport, as opposed to platelet transport, in the present study assumes that C_k is 0, when solving for the concentration of

RBCs, C_j. The reason is that platelets are assumed to not impact the motion of RBCs, while RBCs are assumed to influence the motion of platelets.

The RBC term is predicted to impact the model after scaling Equation 4.3 ($\psi_{platelets}\big|_{wall} \approx C_{platelets}\big|_{wall}\left(1+\phi\big|_{wall}\right)$) at the wall where platelets react with the surface. Based on the scaling, hematocrit does not strongly influence the field potential for fully developed flow where a plasma skimming layer exists, but does become contributory if the near-wall hematocrit levels increase from a flow disturbance. Others have considered the addition of a shear and viscosity term in the field potential [29-30]. However, we do not include a viscosity term because the blood in the present study is considered Newtonian due to the high shear rates in the flow considered here, which would breakup rouleaux. We do not include a shear term because the near-wall shear rates are constant, resulting in a shear rate gradient that is negligible.

Spatial variations in hematocrit were prescribed in Hund and Antaki. Here, we assume that the RBC transport also can be modeled using Equation 4.1. Since there has yet to be an underlying theory that describes and can predict RBC motion, we limit ourselves to the prescription of a drift term. A drift term results in a non-uniform equilibrium concentration for particles. Eckstein and Belgacem used a drift term for platelets to account for platelet margination [31]. Therefore, in the transport of RBCs, Equation 4.2 is augmented with an additional drift term:

Equation 4.4

$$\vec{J}_j = \zeta'_j C_j - \vec{v} C_j - D_j \nabla \psi_j$$

where ζ' is the drift term that is equal to 0 for a uniform equilibrium concentration profile. The conservation of mass is conserved with plasma present in regions that lack platelets or RBCs.

Computational Implementation

We investigate sufficient conditions for experimental thrombus growth rates and extrapolate the model to predict occlusion times of stenoses in idealized coronary arteries by evaluating platelet transport through the convection-diffusion equation using computational fluid dynamics. As a first order approximation, flow is evaluated through axisymmetric stenoses that are defined through geometry based on:

Equation 4.5

$$\frac{r}{r_0} = 1 - \frac{st}{2(100\%)}\left[1 - \cos\left(\pi\left(\frac{x}{x_0} + 1\right)\right)\right]$$

where x_0 is half of the stenosis length, r_0 is the nominal radius, st is the stenosis severity in percent stenosis by diameter, and r is the radius at a given axial coordinate, x. The boundary layer was allowed to develop by providing a straight domain 7 diameters upstream and 15 diameters downstream of the stenosis in the computational domain. The theoretical entrance length for fully developed flow (0.06 Re*D) is below 2 diameters, since our lowest Reynolds number (Re) is 2 in this study. We apply an artificially large diffusivity to platelets and red blood cells in the first 3 diameters of the computational domain to assure concentrations profiles are at equilibrium before entering the stenosis. A long downstream domain is included to allow flow that detaches to reattach. Otherwise errors can occur in the outlet boundary for flow that would normally reenter the boundary in a recirculation region.

Gambit (ANSYS Santa Clara, CA) was used to generate and mesh the stenosis geometry. A structured mesh was created with the largest radial mesh dimension set at $5x10^{-3}$ vessel diameters, which is on the order of a RBC diameter for a 1.5 mm diameter vessel. The axial length of the mesh is $5x10^{-3}$ vessel diameters at x =0, while stretching to a larger dimension away from x=0. The mesh has high resolution radially and near x=0 to capture the boundary layer when a stenosis and thrombus is created. Mesh independence was verified, except for nearly occluded (>95% stenosis) vessels. These very severe stenoses can occur as the thrombus in the simulations approach occlusion and the lumen diameter reduces to the grid size. To keep solutions in the computational domain tractable, we would require an adaptive mesh as thrombus grows to create these very severe stenoses. Since we apply various user-defined functions to the commercial finite volume computational package, Fluent 6.3.26 (ANSYS Santa Clara, CA), our user-defined functions become incompatible with the adaptive meshing capabilities of the software. Therefore we limit the results to less severe stenoses.

Three different computational domains are considered in the present study, including: thrombus, thrombus-border, and normal computational cells. Physiologically, the thrombus domain is a region composed of thrombus, which is assumed to be impenetrable by the fluid. The thrombus-border is a region where platelets are allowed to react with thrombus and deposit on the thrombus surface, while the normal domain depicts a region where blood flows freely. Different components of the governing equations are used, depending on which domain the computational cell exists within. The divisions are depicted in Figure 4.2 and Figure 4.2. The mass and momentum governing equations are solved in all domains with a momentum sink in the thrombus

domain. Thrombus in this study is assumed to act as a solid barrier. Therefore, the initial stenosis and subsequent thrombus is defined as thrombus cells. The momentum sink is defined by applying a very large viscous resistance in the momentum equation and acts as a solid region for the Navier-Stokes equations. The flow field is determined in the normal and border cells through the Navier-Stokes equations since species transport is dependent on the flow. The no-slip condition is applied along the wall.

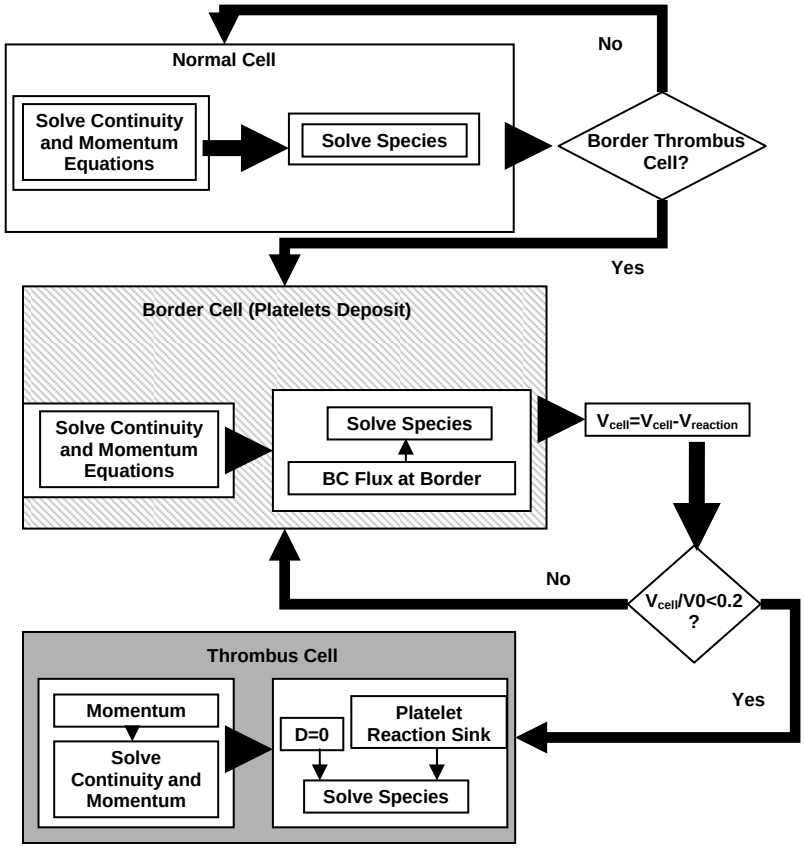


Figure 4.1: Schematic of the methods used to model thrombus growth. Three different domains exist consisting of a normal, thrombus-border, and thrombus computational cell. The flow field is determined from the Navier-Stokes equations in all computational cells. The thrombus cell contains a momentum sink to act as a solid region. Subsequently species transport is determined through the convection-diffusion equation with a sink term in the thrombus with a flux term acting as a boundary condition between a thrombus-border and thrombus cell to simulate deposition of platelets. The convection-diffusion equation is solved for RBCs and platelets. Diffusion and flux are set to 0 in the thrombus domain. After the volume fraction of a thrombus-border cell contains a volume fraction of 0.8 platelets, the cell is converted to a thrombus cell. All computational cells adjacent to a thrombus cell are converted to a thrombus-border cell.

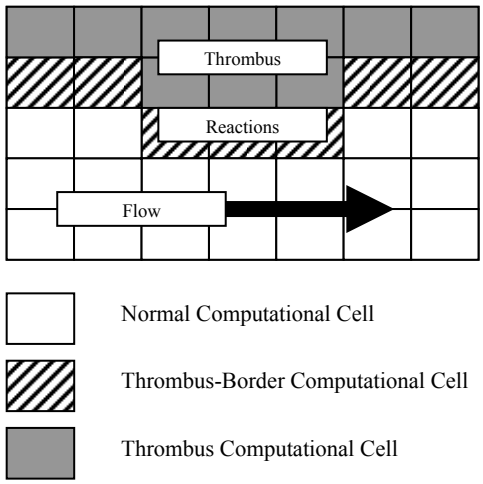


Figure 4.2: Illustration of an example distribution of computational domains. The thrombus occurs at the wall and grows from the wall. Cells bordering thrombus cells are thrombus-border cells and all other cells are normal computational cells.

After Fluent updates flow variables, it solves for the species transport through the convection-diffusion equation, Equation 4.1, as a user-defined scalar in Fluent. In the thrombus domain, the diffusion and convective terms of Equation 4.1 are set to 0. Equation 4.1 is solved in both the normal domain and the thrombus-border domain. The difference between the two domains is that a flux is implemented between the thrombus-border and thrombus:

Equation 4.6

$$J_j = k_t C_j$$

where k_t is the reaction rate of platelets at a thrombus surface. A sink term for the species transport equation in the thrombus domain extracts a species, platelets, from the domain. To conserve mass, the platelets that react in Equation 4.6 are converted to thrombus. Numerical errors resulting from the calculation of gradients to determine platelet flux were smoothed by applying a moving average of flux predicted from Equation 4.6 along the axis of the stenosis. Such averaging is expected to naturally and physically happen in

experimental thrombus growth, since attaching platelets are dynamic and can move along the thrombus surface [32-33]. When the volume fraction of a thrombus-border cell is more than 80% thrombus, it is converted to a thrombus cell. Similarly all normal cells that border the newly defined thrombus cells are converted to thrombus-border cells. 80% is chosen as a thrombus threshold because the platelet packing factor is estimated to be 0.8 [34-35]. As a comparison the maximum atomic packing factor for spheres is ~ 0.74. No flux of platelets or RBCs is allowed at the wall by applying a Neumann boundary condition.

First order upwind discretization schemes are used to solve the Navier-Stokes and convection-diffusion equations. Since sudden changes are temporally made in the flow domain of the simulations due to converting cells into different computational domains, a second order scheme is not used for stability purposes. A coupled method is used for pressure-velocity coupling. Fluid density is set at 1025 kg/m^3 and fluid kinematic viscosity is set at 0.0038 kg/m-s. Flow is assumed laminar and unsteady. Simulation time is required for determining how much thrombus has grown and where it has grown, which is why we implement an unsteady simulation.

Computational Implementation: Diffusivity term

For solving the constitutive equation of diffusion, we use an enhanced diffusivity term, which assumes that the random motion caused by particle interactions is similar in concept to the random motion in thermal diffusivity. Goldsmith has derived that the number of interactions would be proportional to $c_k\dot{\gamma}$ [13]. We use the curve fit from Zydney and Colton [6] to define enhanced diffusivity, which is based off of and earlier model [7], and is expressed as:

Equation 4.7

$$D_e = \alpha\dot{\gamma} + D_{th}$$

$$\alpha = ka^2\phi\left(1-\phi\right)^n$$

where ϕ is the hematocrit, k is a constant at 0.15, and the value for the constant n is 0.8 ± 0.3. The particle diameter, a, is a scale for particle collisions and is taken as the approximate radius of a red blood cell, 4 μm. The thermal diffusivity, D_{th}, is determined through the Stokes-Einstein's equation:

Equation 4.8

$$D_{th} = \frac{k_B T}{6\pi\mu R_{eq}}$$

where μ is the kinematic viscosity of the fluid, k_B is the Boltzmann constant, T is the temperature (310 K), R_{eq} is the equivalent radius of a sphere (≈4.2 μm for a RBC and 1.9 μm for a platelet). Therefore, the thermal diffusivity of a platelet can be approximated as $3.1\text{x}10^{-14}$ cm^2/s and a RBC is $1.4\text{x}10^{-14}$ m^2/s.

Computational Implementation: Drift term

Margination is modeled by including RBCs in the flow domain. Relationships between the fluid flow and RBCs are still not well understood from first principles. As such, the forces that create a non-uniform concentration of RBCs are not known. Therefore we prescribe an equilibrium concentration of RBCs *a priori* by utilizing a drift term, similar to Eckstein and Belgacem [31]. When RBCs are in equilibrium, the flux is 0. The potential for RBC transport is set at the concentration of RBCs, since the motion of platelets is not thought to influence RBC transport due to their small size. In cylindrical fully-developed flow, the drift term can be determined from:

Equation 4.9

$$0 = \zeta'_j - D_j \frac{\partial C_j}{\partial r}$$

Integration results in the drift term:

Equation 4.10

$$\zeta'_j = D_j \frac{C'_{j,eq}}{C_{j,eq}} C_j$$

where $C_{j,eq}$ is the equilibrium RBC concentration profile and $C_{j,eq}$' is the gradient of the RBC equilibrium concentration profile in the radial direction. RBC profiles that are not in equilibrium approach the equilibrium profile at the rate proportional to D_j. The equilibrium profile is updated with the new thrombus boundary as simulations progress and thrombus grows. Hund and Antaki [28] found that the exact type of profile does not strongly influence the transport of platelets. Therefore, we attempt to match the RBC profile of Aarts [15] through a hyperbolic tangent function:

Equation 4.11

$$c_{fd} = H\left(1 + \tanh\left[\alpha\left(r - \delta\right)\right]\right)$$

where H is an arbitrary constant that allows the concentration profile over the domain to be equal to the average concentration, but does not remain in computations due to a cancelation in Equation 4.10. The other constants α and δ are used to shape the hyperbolic tangent and are determined in the results.

Computational Implementation: Hemodynamic Transport Relationship

To determine the ability of the model to predict thrombus growth rates, we compare computed growth rates with experimental growth rates of chapter 2 relative to

shear. Shear is chosen because thrombus growth rates were shown to be correlated with shear in chapter 2 and the enhanced diffusivity in the present study is linearly proportional to shear. Stenosis geometry in the simulation was matched with experimental geometry by performing a least squares fit of the experimental stenosis boundary to the stenosis function in Equation 4.5. The nominal radius of the vessel was set at r_0=0.75 mm with a constant flow rate of 0.25 ml/min applied at the inlet with a parabolic velocity profile. The length of the stenosis, x_0, was found to be approximately $3r_0$. Atmospheric pressure was set as the outlet boundary condition. Transport conditions were investigated by varying the diffusivity parameter. Conditions for the study are given in Table 4.1. Various k_t values were also evaluated to determine the kinetic rate that best fits the experimental data. We assume rigid walls, no pulsatility, and no vessel curvature other than the presence of the stenosis in order to match experimental conditions.

Table 4.1: List of conditions for computational simulations to study transport and binding kinetics

Case	Initial St (%)	Flow Rate (ml/min)	Field Potential	k_t (m/s)	Diffusivity
Transport Computations					
Model	84	0.25	Platelets & RBCs	Inf	Enhanced
Uniform RBC	84	0.25	Platelets	Inf	Enhanced (Hct=0.45)
Thermal Diff	84	0.25	Platelets & RBCs	Inf	Thermal
Analytical	84	0.25	Platelets	Inf	Enhanced
Kinetic Binding Rate Computations					
Model 1 (10^{-3})	84	0.25	Platelets & RBCs	$1x10^{-3}$	Enhanced
Model 2 (10^{-4})	84	0.25	Platelets & RBCs	$1x10^{-4}$	Enhanced
Model 3 (10^{-5})	84	0.25	Platelets & RBCs	$1x10^{-5}$	Enhanced
Analytical	84	0.25	Platelets & RBCs	$1x10^{-4}$	Enhanced

Computational Implementation: Occlusion Time

Thrombus occlusion times are computed by applying the model to pathophysiological conditions. Thrombus is allowed to grow until a 95% stenosis, at which occlusion is defined to occur due to a mesh resolution limitation. Geometry for the pathophysiological model scaled with the vessel size. A vessel diameter of 3 mm was used to match the size of a idealized coronary artery. We used a stenosis length of x_0=3 r_0 and applied Equation 4.5 to obtain stenosis geometry. Varying degree stenoses are evaluated, as shown in Table 4.2. An inlet flow rate was determined by the relationship between stenosis severity at a given moment in time and the average flow rate from Gould et al. [36]. Gould et al. studied how reactive hyperemia can maintain a relatively constant flow rate until ~75% stenosis severity, after which there was little data. A computational study was performed to evaluate how flow rates would change for very severe stenoses, resulting in the relationship:

Equation 4.12

$$\bar{Q} = \begin{cases} Q_0 & st<74\% \\ Q_0\left(0.0015st^2 - 0.3st + 15\right) & st>74\% \end{cases}$$

where Q_0 is the average flow rate of the artery under physiological conditions in the absence of a stenosis. We neglect the pulsatility of the flow and assume that the thrombus growth averaged over time for pulsatile conditions will be approximately the same as thrombus growth based on the average flow rate through the vessel. Pulsatility has only secondary effects on the shear and mass transport over a stenosis apex in the coronary artery [37-39]. Furthermore we assume that non-rigid vessel walls and vessel curvature only have secondary effects on thrombus growth. The maximum flow rate in

this study results in a Re of 230, which is the lower bound critical upstream Re for reaching the transition to turbulence in the region of a stenosis [40]. Therefore, we assume laminar flow throughout the simulation. Atmospheric pressure is applied as an outlet boundary condition.

Table 4.2: List of conditions for computational simulations to study occlusion time

Case	Initial St (%)	Flow Rate (ml/min)	Field Potential	k_t (m/s)	Diffusivity
Occlusion Time Computations					
Occlusion (40)	40	Equation	Platelets & RBCs	$1x10^{-4}$	Enhanced
Occlusion (50)	50	Equation	Platelets & RBCs	$1x10^{-4}$	Enhanced
Occlusion (60)	60	Equation	Platelets & RBCs	$1x10^{-4}$	Enhanced
Occlusion (70)	70	Equation	Platelets & RBCs	$1x10^{-4}$	Enhanced
Occlusion (80)	80	Equation	Platelets & RBCs	$1x10^{-4}$	Enhanced

Analytical Model

The model describing platelet motion, Equation 4.1, is simplified to develop an analytical expression for platelet deposition rates based on transport for estimating the rate of thrombus growth. The analytical model consists of flow through an axisymmetric straight cylindrical vessel. Since thrombus grows at a maximum of approximately 2 platelets/μm^2-min in chapter 3 and platelets enter a coronary artery at a rate that is an order of magnitude higher, 64 platelet/μm^2-min ($J_{in}=QC_{inf}/A$), the unsteady term in the convective-diffusion equation can be neglected. The calculation assumes the flow rate, Q, is 2 ml/s, the bulk concentration of platelets, C_{inf}, is 225,000 platelets/μl, and a vessel diameter of 3 mm. Flow is assumed to be fully developed. Axial convection is assumed to dominate over axial diffusion. Additionally, the field potential is assumed to be equal to the platelet concentration. Therefore, the convective-diffusion equation simplifies to:

Equation 4.13

$$v_z \frac{\partial C_j}{\partial z} = D_j \left(\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial C_j}{\partial r} \right) \right)$$

where v_z is the axial velocity, r is the radial coordinate, and z is the axial coordinate. Near the wall, we approximate the velocity as $\dot{\gamma}_w$ y, where y is the normal distance from the wall and $\dot{\gamma}_w$ is the wall shear rate. Thus we use the Lévêque approximation. This near-wall approximation assumes a linear velocity gradient. Coordinates are converted to a distance from the wall, y, which is assumed to be much smaller than the radius. To solve for the new equation we can use the similarity variable like the one from Turitto et al. [23]:

Equation 4.14

$$\eta = \left(\left(\frac{\dot{\gamma}}{D_j z} \right)^{1/3} y \right)$$

We solve for concentration with respect to the similarity variable:

Equation 4.15

$$C_j = A \int_0^{\eta} e^{-\frac{1}{9}\eta^3} d\eta + B$$

We now require boundary conditions for mass transport. We assume that margination results in platelet concentrations at the wall that are a factor of 2 greater than bulk flow concentrations in the analytical model [25]. We use Dirichlet and Neumann boundary conditions to solve Equation 4.15:

Equation 4.16

$$C_j\Big|_{\eta\to\infty} = 2C_{j,\inf}$$

$$J_{\eta\to 0} = D\frac{\partial C_j}{\partial \eta}\frac{\partial \eta}{\partial y}\Bigg|_{\eta\to 0} = k_t C_j\Big|_{\eta\to 0}$$

where k_t is the kinetic binding rate constant of a platelet to the thrombus site. The resulting flux of platelets predicted to arrive at the surface is:

Equation 4.17

$$J_j = \frac{2\left(C_{j,\inf}\right)}{\frac{1}{k_t} + \frac{\left(1.8575\right)}{\left(\frac{\dot{\gamma}D_j^2}{z}\right)^{1/3}}}$$

The axial dimension, z, is set at $3R_0$ in the present study to match x_0 of Equation 4.5.

Results

RBC and Platelet Concentration Distribution

Since the platelet and RBC distributions are important to transport, we evaluate the concentration profiles for flow through a straight axisymmetric vessel, as presented in Figure 4.3. The hematocrit, or RBC volume fraction, is highest near the center of the vessel, with a hematocrit of 0.57, while near-wall hematocrit has a value of 0.017. Conversely, the lowest platelet volume fraction of 0.0012 is at the center of the vessel, while the highest volume fraction of 0.025 is at wall.

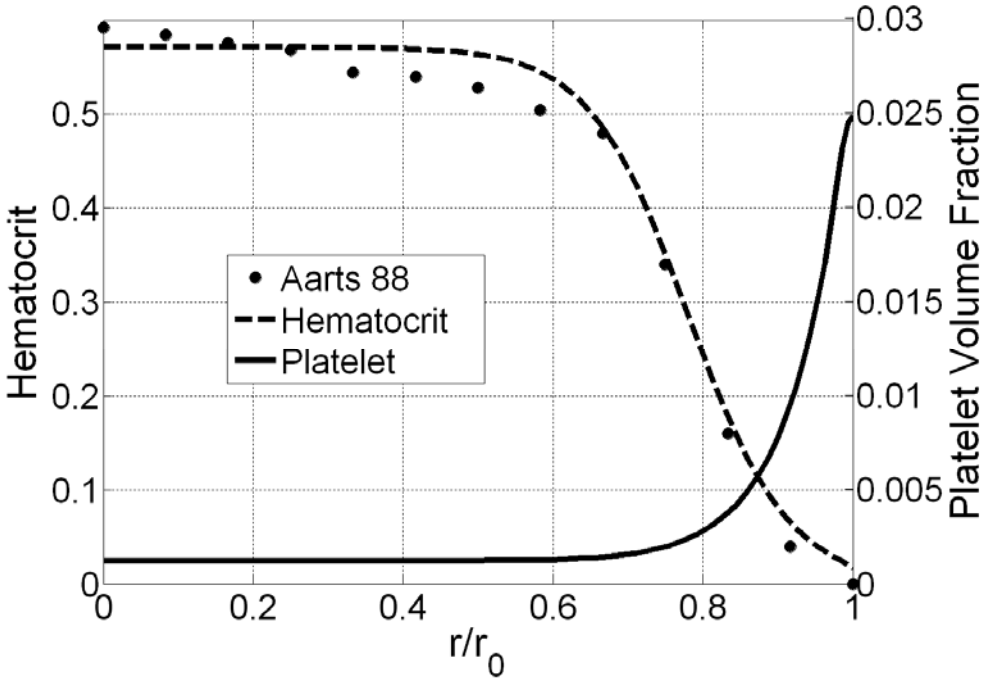


Figure 4.3: Radial concentration profiles of a RBCs and platelets. A drift term was added to the convection-diffusion equation to account for the non-uniform RBC concentration profile. The equilibrium profile was set to match an example RBC profile from the Aarts et al. [15]. To match the profile, α was set to $7.8/r_0$ and δ was set to $0.23\ r_0$ in Equation 4.11, where r_0 is the radial location of the nearest wall or thrombus border. The platelet concentration profile has a maximum of ~$15C_{inf}$. The peak of the platelet concentration profile matches well with Zhao et al. [14].

Two different flow conditions were considered in the study. The first condition, consisting of flow over a 84% stenosis with a upstream Re of 2, results in a parabolic velocity profile at the stenosis apex and no flow separation, with converging flow near the apex of the stenosis, Figure 4.4A,B. The model hematocrit concentration profile is fully developed before entering the stenosis region, as can be seen in Figure 4.4C where there is a plasma skimming layer near the wall, corresponding to a region of low hematocrit. The hematocrit profile matches Figure 4.3 in this region. Similarly, the platelet profile is fully developed upstream of the stenosis, Figure 4.4D. The species transport model results in hematocrit and platelets that diffuse slightly as they enter the

stenosis. The diffuse characteristics remain downstream of the stenosis apex with values close to the average radial values. The area-average hematocrit is higher at the apex relative to any other location in the computational domain, but the mass average, which accounts for the velocity is the same since mass of RBCs is conserved. A high area-average increases the RBC gradient near the wall, which increases the field potential defined in the current study. Concentration profiles in the downstream region continue to develop back to their equilibrium profile. Therefore, the densest region of platelets is upstream of the stenosis.

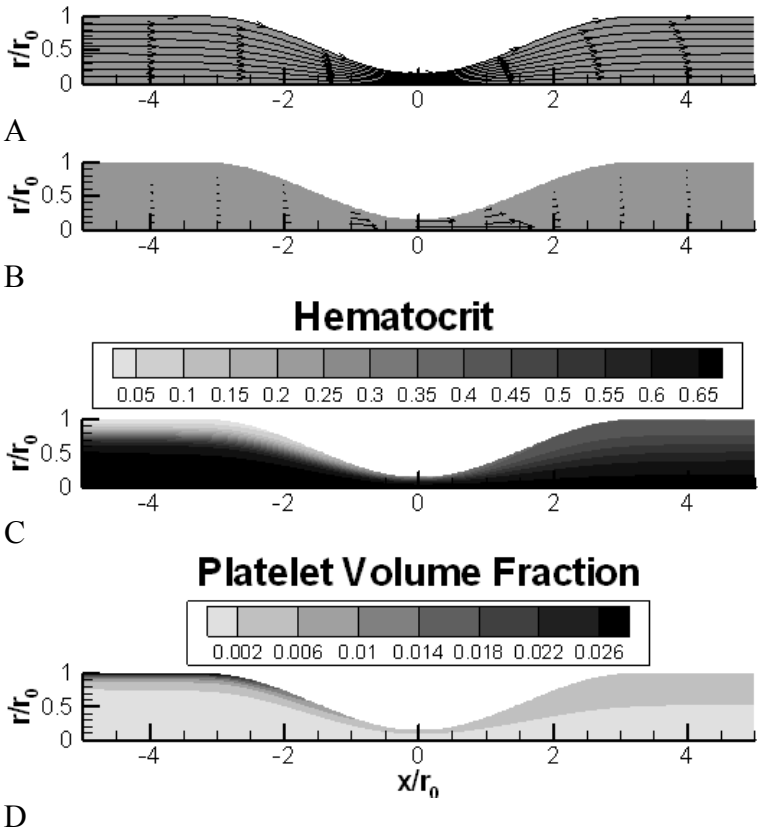


Figure 4.4: Flow computed from the 84% stenosis model A) Streamlines that converge at the stenosis region. B) Velocity vectors. The velocity profile at the stenosis apex remains parabolic. C) Hematocrit contours with a high hematocrit near the center of the vessel and a plasma skimming layer near the wall. The stenosis distributes the RBCs more uniformly along the vessel. D) Platelet volume fraction contours with a high concentration near the wall due to margination. Similar to RBCs, the platelets distribute more uniformly near the apex of the stenosis relative to the upstream region. Flow computed from a 60% stenosis by diameter model

Coronary artery flow conditions were also considered with an upstream Re of 230 and a stenosis of 60%. Flow separates slightly downstream of the stenosis apex, Figure 4.5A,B. Otherwise, flow characteristics remain similar to the low Re case. Again hematocrit is the highest near the core of the vessel upstream of the stenosis, with high platelet concentrations near the wall, Figure 4.5C,D. The magnitude of maximum hematocrit and platelet volume fraction are reduced relative to the low Re case. The recirculation region is relatively void of RBCS compared to the core flow in the downstream section, as opposed to the low Re case where hematocrit is spread in a uniform-like manner.

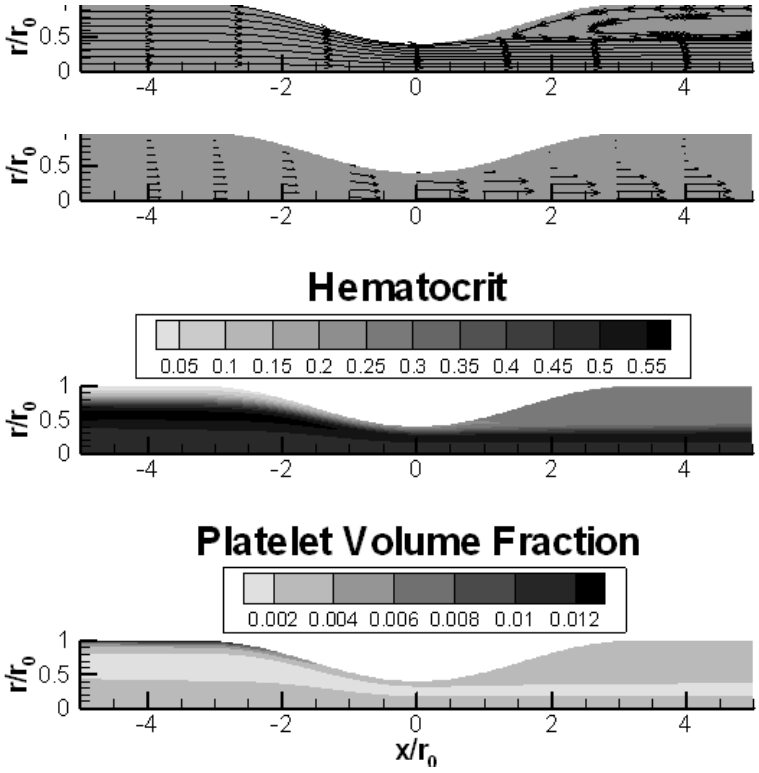


Figure 4.5: A) Streamlines that converge at the stenosis region, with flow separation near the stenosis apex, illustrated by a recirculation region distal to the stenosis. B) Velocity vectors. The velocity profile at the stenosis apex becomes blunt. There is a jet-like flow exiting the stenotic region. C) Hematocrit contours with a high hematocrit near the center of the vessel and a plasma skimming layer near the wall and a more uniform distribution of RBCs at the apex. There are relatively few RBCs that enter the recirculation region, with the highest hematocrit remaining near the vessel core. D) Platelet volume fraction contours with a high concentration near the wall due to margination and a more uniform distribution near the apex of the stenosis.

Hemodynamic Relationship to Thrombus Growth Rates Based on Platelet Transport

Since thrombus growth rates have been shown to be correlated with wall shear rates experimentally, we investigate the thrombus growth rates predicted from the species transport model relative to shear. First, a kinetic binding constant, associated with platelet to surface binding, of $k_t=1x10^{-3}$ m/s is evaluated in the species transport model with a RBC dependent field potential to simulate a transport limited process. The results were the same with higher kinetic binding rates for shear rates from 100 to 10,000 s^{-1}. The model boundary conditions (upstream Re=2, st=84%, R_0=0.75 mm) match those of the experiments from Chapter 3. Model conditions are given in Table 4.1. Thrombus growth rates increase monotonically with shear, similar to the experiments, Figure 4.6. The model falls within 1 experimental standard deviation.

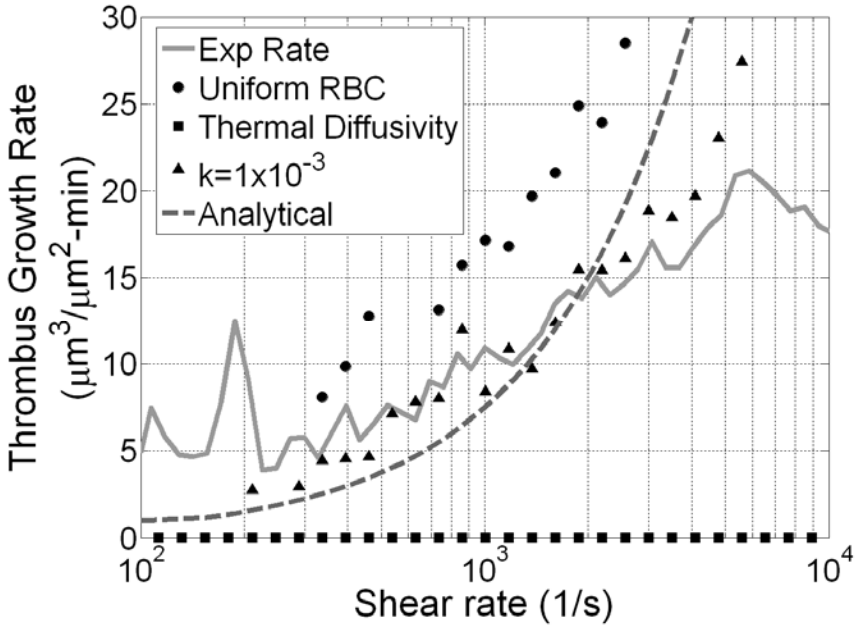


Figure 4.6: Thrombus growth rates relative to shear rates for different transport conditions. The species transport model with a RBC term in the field potential denoted by its binding rate, $k_t=10^{-3}$ m/s, which is effectively an infinite binding rate for the shear rates plotted here, matches within a standard deviation of the experimental data. The uniform RBC distribution results exceed experimental thrombus growth rates at shear rates nearing 10^4 s^{-1}. Platelets transporting at the rate of thermal diffusivity has a growth rate that underpredicts thrombus growth rates. The analytical model from Equation 4.17 also lies within a standard deviation of the experimental data. An infinite kinetic binding rate was assumed for the analytical model.

We further verify that thrombus growth rates predicted from transport rate-limited processes follow the same trend and are on the same order of magnitude as the experimental growth rates at shear below 10,000 s^{-1}, by considering additional models. First, the species transport model is investigated with a field potential (C_i=0 in Equation 4.3) that only consists of platelets. The model, therefore, does not account for margination. This type of model is historically more common in the literature. Again, thrombus growth rates are predicted to increase relative to shear, Figure 4.6. Thrombus growth rates predicted from this model exceed experimental values and begin to exceed 1 standard deviation of the experimental values near a shear rate of 10,000 s^{-1}. The model

consists of higher near wall platelet diffusion, but without the high near-wall concentration of platelets associated with margination.

Predictions of thrombus growth rates with the analytical model, Equation 4.17, are also investigated since the assumptions of the analytical model are met with Stokes-like flow that exists throughout the flow domain with the experimental boundary conditions. The analytical model results in a similar growth rate trend with respect to shear rate and also predicts thrombus growth rates within a standard deviation of the experimental growth rates, Figure 4.6.

The requirement for platelet enhanced diffusivity is studied by only considering thermal diffusivity for platelets. Model predictions based off of thermal diffusivity underpredict thrombus growth rates by orders of magnitude, Figure 4.6. Thrombus growth is predicted to take a month to occlude a vessel if only platelet thermal diffusivity is considered. Occlusion based only on thermal diffusivity also occurs in a uniform manner across the entire stenosis, instead of localizing at the apex, which does not agree with the experimental results. The experimental results, instead, consist of significant thrombus near the stenosis apex, with little thrombus elsewhere.

The model predicts increases in thrombus growth rates relative to shear rates without bound. Thrombus growth rates predicted in the model can be bounded at high shear rates if a kinetic binding parameter is included as a boundary condition at the wall. Therefore, we consider kinetic binding rates that vary by three orders of magnitude to approximate the binding rate that best matches experimental data. The conditions for the computations are given in Table 4.1. For $k_t=1x10^{-3}$ m/s and $k_t=1x10^{-4}$ m/s, thrombus growth rates increase relative to shear at similar rates as an infinite platelet to surface

kinetic binding rate for shear below 6,000 s^{-1}, as illustrated in Figure 4.7. However, a lower kinetic binding rate (k_t=1x10^{-5} m/s) results in thrombus growth rates that are an order of magnitude lower than experiments. A kinetic binding rate of k_t=1x10^{-4} m/s results in the closest maximum thrombus growth rates to experimental thrombus growth rates with the most accurate predictions for shear rates exceeding 6, 000 s^{-1}. The analytical model of Equation 4.17, using a kinetic binding rate, k_t=1x10^{-4} m/s, predicts a similar trend of thrombus growth rates as the numerical species transport model.

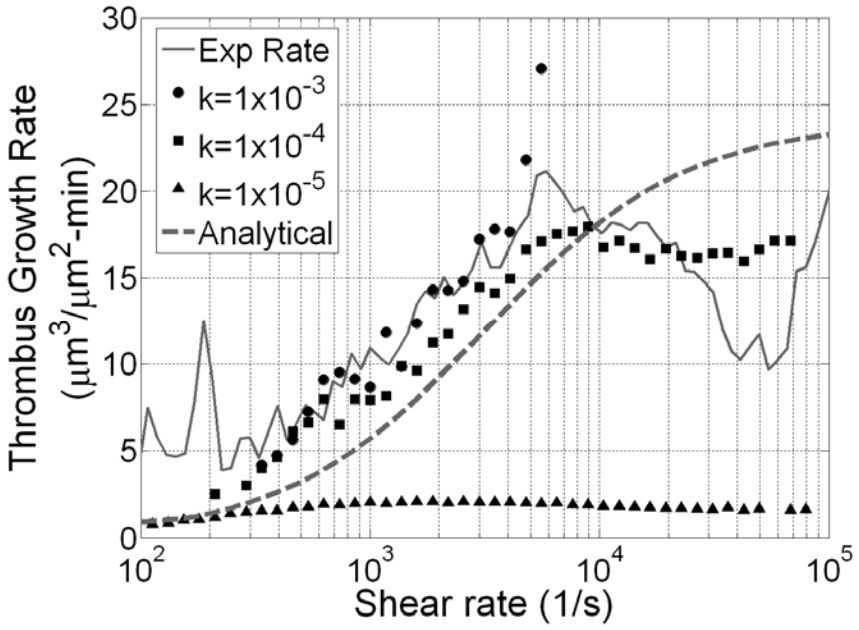


Figure 4.7: Thrombus growth rates relative to shear with different kinetic binding rates including: k_t=10^{-3} m/s, k_t=10^{-4} m/s, and k_t=10^{-5} m/s. The kinetic binding rate, k_t=10^{-4} m/s, fit the experimental data the best. The analytical model from Equation 4.17 is also plotted with a kinetic binding rate, k_t=10^{-4} m/s.

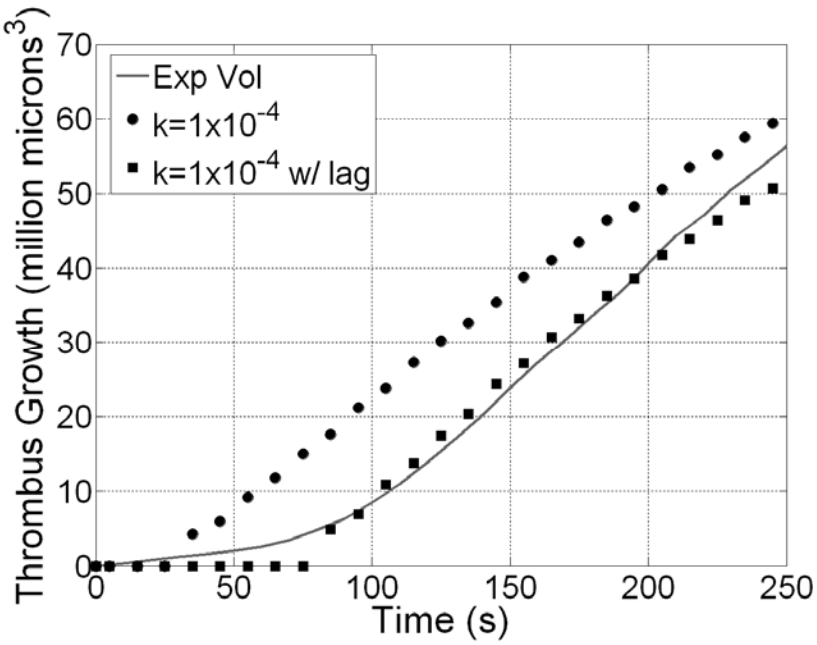
Thrombus volume growth is also evaluated with respect to time as a comparison with experiments, to verify that the model is properly predicting thrombus growth. A kinetic binding parameter of k_t=1x10^{-4} m/s fit the experimental data the best. Therefore, we evaluate thrombus growth over time using this kinetic binding rate. The results show

a monotonically increasing function with respect to time, Figure 4.8A. Volume growth predictions are temporally offset from the experimental volume growth. However, experimental volume growth exhibited a lag time before thrombus began growing. The experimental regression analysis illustrated a power-law relationship between lag time and shear. Therefore, we apply the relationship to same power to impose a lag period for thrombus growth to the model, resulting in:

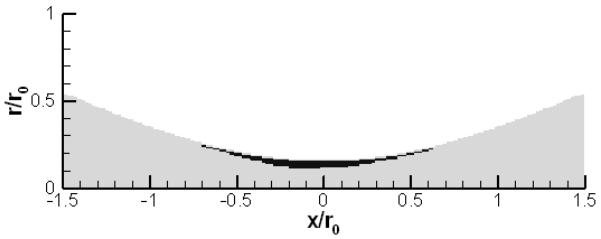
Equation 4.18

$$t_{\text{lag time}} = 280\dot{\gamma}^{-0.2}$$

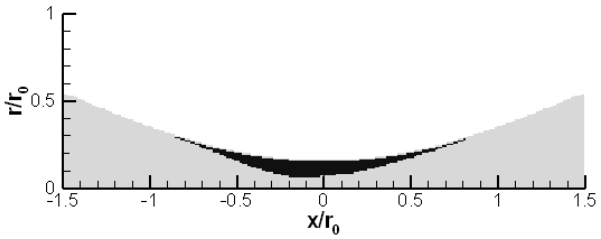
The experimental lag time had a coefficient of 718 instead of 280, but was also based on 10 μm of previously deposited thrombus, which would add to the lag time. Therefore, we used a value in the model that allowed us to estimate volume growth more closely. Thrombus grows near the throat of the stenosis, as illustrated in Figure 4.8B,C. The location of growth, the quantity of growth, and the growth rate predicted by the current model all match well with experiments.



A



B

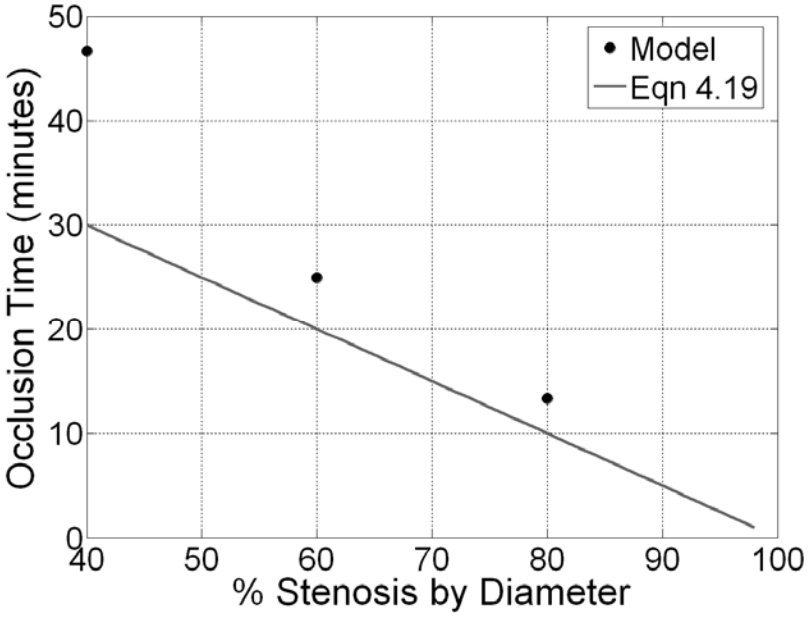


C

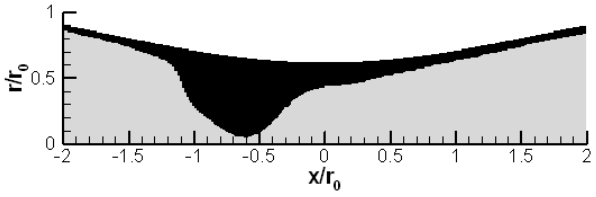
Figure 4.8: A) Computed thrombus volume growth and experimental growth relative to time for $k_t=10^{-4}$ m/s. Thrombus formed over a 84% stenosis for $k_t=10^{-4}$ m/s is illustrated after a time of B) 2 minutes C) 4 minutes.

Vessel Occlusion Time from Thrombus Growth Rates in the Presence of a Pathophysiological Stenosis

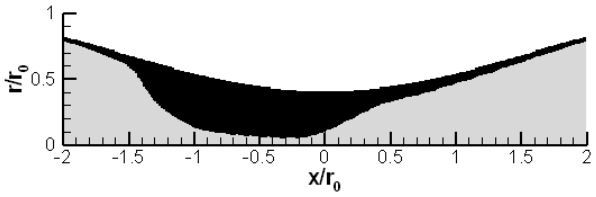
Thrombus growth creates a progressive stenosis in a vessel until the vessel is fully occluded. To determine the time to reach occlusion, the model was extended to evaluate thrombus growth over stenoses of varying severity, Table 4.1. A constant kinetic binding rate of $k_t=1x10^{-4}$ m/s was used to evaluate dynamically growing thrombus. Occlusion time predicted from the model is a monotonically decreasing function with respect to stenosis severity, Figure 4.9A. Occlusion times decrease monotonically as stenosis severity increases. Thrombus grows upstream of the stenosis throat and at the stenosis apex for pathophysiological conditions. The kinetic binding rate limits the rate of thrombus growth whenever shear exceeds 6,000 s^{-1}. Final growth for a 40%, 60%, and 80% stenosis is depicted in Figure 4.9B,C,D, to illustrate the uniform growth. Little growth is seen distal to the stenosis apex. Flow separates in the distal region and shear rates remain low in the recirculation region. Low shear rates result in an enhanced diffusivity term that approaches the thermal diffusivity value. The initial 40% stenosis results in thrombus formation that forms in a narrower region of the stenosis. Initial shear rates for the 40% stenosis are lower than 6,000 s^{-1}, allowing transport to play a larger role in where thrombus emanates. As thrombus grows, it does exceed 6,000 s^{-1} and continues a rate based on the kinetic binding constant. The other stenoses start out at higher shear rates.



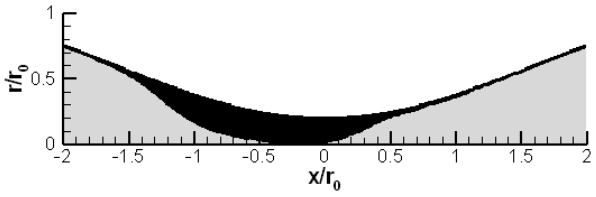
A



B



C



D

Figure 4.9: Occlusion time for a given stenosis. The occlusion time decreases as the stenosis severity increases. A) The occlusion time is plotted relative to stenosis severity. B) Thrombus growth of an initial 40% stenosis after 47 minutes. C) Thrombus growth of an initial 60% stenosis after 25 minutes. D) Thrombus growth of a initial 80% stenosis after 13 minutes.

Since average shear rates remain above 6,000 s^{-1} near the stenosis apex for stenoses exceeding 40%, we can assume that thrombus will grow at a constant rate of $J_{thrombus} = k_t C_{platelets}\big|_{wall}$. The platelet volume fraction at the wall for kinetic rate-limited growth consistently resulted in a value of approximately $C_{platelets}\big|_{wall} = 0.004$. Since we assume that platelets have a volume fraction of 0.8 in thrombus, we can predict the time to vessel occlusion for a given stenosis severity:

Equation 4.19

$$t_{occlusion} = \frac{0.8\left(1 - st/100\%\right) r_0}{k_t C_{platelets}\big|_{wall}}$$

Results of this equation are also included in Figure 4.9A. The equation predicts occlusion times that are slightly faster than species transport model predictions. Since, the initial shear rates of the 40% stenosis are below the kinetic-rate limiting threshold determined in this study, there is an initial time where transport rate-limited thrombus growth occurs, accounting for the larger mismatch between Equation 4.19 and model predictions for a 40% stenosis.

Discussion

The species transport model was used to quantify thrombus growth rates relative to shear. For shear rates below 6,000 s^{-1}, the model fell within a standard deviation of experimental data. These results are indicative of a transport-rate limited thrombus growth. Based on the present study, decreases in arrival rate of platelets to an atherosclerotic lesion or cardiovascular medical device would decrease thrombus growth rates for shear rates below 6,000 s^{-1}. However, thrombosis is typically targeted by

anticoagulant and anti-platelet drugs that affect platelet binding as opposed to transport, even at these shear rates. Lowering platelet arrival rates has the potential as a therapy that still allows normal hemostasis in the body, while minimizing thrombus risk on a target area based on the present work.

Lowering thrombus growth rates would increase the time to vessel occlusion for an atherosclerotic lesion. It has been previously proposed that there is a lower risk for complete thrombus occlusion when the time to occlusion increases [25]. Specifically stenoses with occlusion times that takes longer than an hour may not ever occlude based on the low probability of occlusion for the same stenoses in experimental studies, while occlusion that occurs within 30 minutes has a higher probability of fully occluding in experimental studies [41]. If thrombosis has a high probability of occlusion for occlusion times that are less than 30 minutes, then Equation 4.19 predicts that stenoses more severe than 40% by diameter have a high probability of occluding. The species transport model predicts that stenoses more severe than 60% have a high probability of occluding with occlusion times that are faster than 25 minutes. However, occlusion times predicted by the model for a 40% stenosis are greater than 30 minutes and are near an hour, indicating a low probability of occlusion. Thrombus for the 40% stenosis also formed with a different, narrow, geometric shape due to the initial transport rate-limited growth. The shape may play a role in whether or not thrombus occlusion can occur because a narrow shape may be more likely to emobolize due to high stresses in the thrombus structure.

Equation 4.19, used as a rough estimate for predicting the risk of thrombus occlusion of a stenosis, matches well with Wootton et al. who reports occlusion times for a 70% and 50% stenosis are 15 minutes and 27 minutes respectively. Equation 4.19

predicts occlusion times of 15 and 25 minutes respectively. Model predictions resulted in slower occlusion times overall. The 80% stenosis occluded in 13 minutes, while a 60% stenosis effectively occluded a vessel in 25 minutes. The 40% stenosis took 47 minutes to occlude. The advantage to utilizing Equation 4.19 is that the only input variables required is the stenosis severity and nominal vessel diameter, which is commonly found in angiograms

There remains a question about whether platelets near a vessel wall experience enhanced diffusivity or thermal diffusivity. Species transport predictions of thrombosis in the absence of enhanced diffusivity results in growth rates that are orders of magnitude lower than experimental growth rates, resulting in occlusion that occurs over a month as opposed to minutes seen in experiments [34]. Therefore, there would need to be a mechanism in the blood that enhances the transport of platelets to a thrombus site. While the present study does not prove that there is an enhanced diffusivity near the wall since this work consists of a model, it does suggest that enhanced platelet diffusivity near the wall must exist for the high thrombus growth rates that can be found in experiments.

As shear rates exceed 6,000 s^{-1}, the model predicts thrombus growth rates that exceed values from the experimental data. This is because the model with an infinite kinetic binding rate increases without bound due to the increases in diffusivity. Thrombus growth rates, however, can be limited by the kinetic binding rate of platelets to a thrombus site. The limited growth was illustrated when various kinetic parameters were investigated. We found that a kinetic parameter of 10^{-4} to fit the experimental data most accurately. This value can be compared with a value of $5x10^{-5}$ m/s found from Wootton et al. and Sorenson et al. [25, 42] and 10^{-4} m/s from Jordana et al. [26] for initial

platelet deposition. Goodman et al. found a kinetic rate constant of $3.5x10^{-3}$ m/s for activated platelets and a value of $2.5x10^{-6}$ m/s for platelets that have not activated. Our kinetic parameter was fit to data of platelets depositing on thrombus, which is assumed to be composed of activated and non-activated platelets. Initial platelet deposition would likely occur on non-activated platelets since platelets do not have time activate on a stenosis, without attaching to the stenosis first [25, 43]. Therefore, our value is between activated and non-activated platelets binding rates found in the literature.

We would also like to recognize that transport could also still limit the rate of platelet deposition at high shear. Enhanced diffusivity determined from Zydney and Colton [6] was based on a curve that was fit to data that consisted of shear rates that only reached 5,000 s^{-1}. It is possible that the transport of platelets and RBCs do not continue to increase with shear. If transport did continue to increase without bound, then it would effectively go to infinite as shear increases, which is a physical impossibility.

The ability of the model to predict thrombus growth was further tested by comparing predicted volume growth over time. Shear rates are continually updated and fed back into the model to determine transport and binding rates. This dynamic process predicted volume growth quantities within a standard deviation of experimental values, except for a region of time near 75 seconds, once a lag time was added to the growth. The location of thrombus growth also matched well with experiments.

This study has several limitations. Enhanced diffusivity for the transport model is based off of data of shear rates below 5,000 s^{-1}, but we extrapolate the function for all shear rates. The grid size as thrombus nears occlusion nears the scale of the lumen at the stenosis apex. The study of thrombus occlusion time is therefore based off of the time to

a 95% stenosis. Due to the low resolution grid spacing of the 95% stenosis, the hemodynamics are not predicted as accurately as less severe stenoses. Thrombus occlusion predictions also neglect many hemodynamic factors that are considered to have a secondary effect on transport, such as pulsatility, rigid walls, and vessel curvature. After a 95% stenosis Fahraeus-Lindquist effects may begin to influence transport and elastic stenoses may potentially collapse from low Bernoulli pressures near the stenosis apex [44-45]. Predictions of thrombus growth rates from species transport assume a constant kinetic parameter that best fits the experimental data. However, there was no significant trend for the experimental data at shear rates exceeding 6,000 s^{-1}, so the model may not be accurate at very high shear rates. Embolization was also not accounted for in the model. Instead platelets that attached to the surface were assumed to remain on the surface. The present study only considers platelet deposition kinetics, as opposed to direct binding kinetics associated with the glycoproteins found on a platelet.

Conclusion

Thrombus growth rates are found to be transport rate-limited for shear rates below 6,000 s^{-1} based on agreement between a species transport model and an experimental model of thrombus growth over a collagen coated stenotic vessel. In order to capture experimental thrombus growth rates, an enhanced diffusivity was required. Thrombus growth predicted to transport from thermal diffusivity, alone, was found to take a month to occlude an 84% stenosis that experimentally took minutes to occlude. An extension of the species transport model can be used to predict thrombus growth over atherosclerosis or medical device.

References

1. Yutani, C., et al., *Coronary atherosclerosis and interventions: pathological sequences and restenosis.* Pathology International, 1999. **49**(4): p. 273-290.
2. Viles-Gonzalez, J.F., V. Fuster, and J.J. Badimon, *Atherothrombosis: a widespread disease with unpredictable and life-threatening consequences.* European heart journal., 2004. **25**(14): p. 1197-1207.
3. Davies, M.J. and A. Thomas, *Thrombosis and acute coronary-artery lesions in sudden cardiac ischemic death.* N Engl J Med, 1984. **310**(18): p. 1137-40.
4. Chandler, A., et al., *Coronary thrombosis in myocardial infarction:: Report of a workshop on the role of coronary thrombosis in the pathogenesis of acute myocardial infarction.* The American Journal Of Cardiology, 1974. **34**(7): p. 823-833.
5. Cadroy, Y., T.A. Horbett, and S.R. Hanson, *Discrimination between platelet-mediated and coagulation-mediated mechanisms in a model of complex thrombus formation in vivo.* The Journal of laboratory and clinical medicine., 1989. **113**(4): p. 436-448.
6. Zydney, A.L. and C.K. Colton, *Augmented solute transport in the shear flow of a concentrated suspension.* PhysicoChemical Hydrodynamics, 1988. **10**(1): p. 77-96.
7. Wang, N.-H.L. and K.H. Keller, *Augmented transport of extracellular solutes in concentrated erythrocyte suspensions in couette flow.* Journal of Colloid and Interface Science, 1985. **103**(1): p. 210-225.
8. Keller, K.H., *Effect of fluid shear on mass transport in flowing blood.* Federation proceedings., 1971. **30**(5): p. 1591-1599.
9. Goldsmith, H.L. and J.C. Marlow, *Flow behavior of Erythrocytes. II. Particle motions in concentrated suspensions of ghost cells.* Journal of Colloid and Interface Science, 1979. **71**(2): p. 383-407.
10. Eckstein, E., D. Bailey, and A. Shapiro, *Self-diffusion of particles in shear flow of a suspension.* Journal of Fluid Mechanics, 1977. **79**(01): p. 191-208.
11. Leighton, D. and A. Acrivos, *The shear-induced migration of particles in concentrated suspensions.* Journal of Fluid Mechanics, 1987. **181**: p. 415-439.
12. Leighton, D. and A. Acrivos, *Measurement of shear-induced self-diffusion in concentrated suspensions of spheres.* Journal of Fluid Mechanics, 1987. **177**: p. 109-131.
13. Goldsmith, H. and S. Mason, *The flow of suspensions through tubes. III. Collisions of small uniform spheres.* Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 1964. **282**(1391): p. 569-591.
14. Zhao, R., M.V. Kameneva, and J.F. Antaki, *Investigation of platelet margination phenomena at elevated shear stress.* Biorheology, 2007. **44**(3): p. 161-177.
15. Aarts, P.A., et al., *Blood platelets are concentrated near the wall and red blood cells, in the center in flowing blood.* Arteriosclerosis : an official journal of the American Heart Association, Inc., 1988. **8**(6): p. 819-824.

16. Bilsker, D.L., et al., *A freeze-capture method for the study of platelet-sized particle distributions.* Biorheology, 1989. **26**(6): p. 1031-1040.
17. MacMeccan, R., et al., *Simulating deformable particle suspensions using a coupled lattice-Boltzmann and finite-element method.* Journal of Fluid Mechanics, 2008. **618**: p. 13-39.
18. Ding, E.J. and C.K. Aidun, *Cluster size distribution and scaling for spherical particles and red blood cells in pressure-driven flows at small Reynolds number.* Physical Review Letters, 2006. **96**(20): p. 204502-1.
19. Liu, Y. and W. Liu, *Rheology of red blood cell aggregation by computer simulation.* Journal of Computational Physics, 2006. **220**(1): p. 139-154.
20. Jung, J., A. Hassanein, and R. Lyczkowski, *Hemodynamic computation using multiphase flow dynamics in a right coronary artery.* Annals of biomedical engineering, 2006. **34**(3): p. 393-407.
21. Fiechter, J., *Numerical study of platelet transport in flowing blood.*, in *Mechanical Engineering*. 1998, Georgia Institute of Technology: Atlanta. p. 76.
22. Bark, D., *Mechanicstic Numerical Study of Thrombus Growth*, in *Mechanical Engineering*. 2007, Georgia Institute of Technology: Atlanta. p. 183.
23. Turitto, V.T. and H.R. Baumgartner, *Platelet deposition on subendothelium exposed to flowing blood: mathematical analysis of physical parameters.* Transactions - American Society for Artificial Internal Organs., 1975. **21**: p. 593-601.
24. Sorensen, E.N., et al., *Computational Simulation of Platelet Deposition and Activation: I. Model Development and Properties.* Annals of Biomedical Engineering, 1999. **27**(4-12): p. 436-448.
25. Wootton, D.M., et al., *A mechanistic model of acute platelet accumulation in thrombogenic stenoses.* Annals of biomedical engineering., 2001. **29**(4): p. 321-329.
26. Jordana, A., et al., *The effects of margination and red cell augmented platelet diffusivity on platelet adhesion in complex flow.* Biorheology, 2004. **41**(5): p. 641-653.
27. Goodman, P.D., et al., *Computational model of device-induced thrombosis and thromboembolism.* Annals Of Biomedical Engineering, 2005. **33**(6): p. 780-797.
28. Hund, S. and J. Antaki, *An extended convection diffusion model for red blood cell-enhanced transport of thrombocytes and leukocytes.* Physics in Medicine and Biology, 2009. **54**: p. 6415.
29. Phillips, R., et al., *A constitutive equation for concentrated suspensions that accounts for shear induced particle migration.* Physics of Fluids A: Fluid Dynamics, 1992. **4**: p. 30.
30. Kao, S., *Platelet Transport and Surface Reactions in Mural Thrombosis*. 2000, Rice University: Houston, TX.
31. Eckstein, E.C. and F. Belgacem, *Model of platelet transport in flowing blood with drift and diffusion terms.* Biophysical journal., 1991. **60**(1): p. 53-69.
32. Ruggeri, Z.M., et al., *Activation-independent platelet adhesion and aggregation under elevated shear stress.* Blood, 2006. **108**(6): p. 1903-1910.

33. Yago, T., et al., *Platelet glycoprotein Ibalpha forms catch bonds with human WT vWF but not with type 2B von Willebrand disease vWF.* The Journal Of Clinical Investigation, 2008. **118**(9): p. 3195-3207.
34. Ku, D.N. and C.J. Flannery, *Development of a flow-through system to create occluding thrombus.* Biorheology, 2007. **44**(4): p. 273-284.
35. Wootton, D.M., *Mechanistic Modeling of Occlusive Arterial Thrombosis*, in *Mechanical Engineering*. 1998, Georgia Institute of Technology: Atlanta. p. 421.
36. Gould, K.L., K. Lipscomb, and G.W. Hamilton, *Physiologic basis for assessing critical coronary stenosis. Instantaneous flow response and regional distribution during coronary hyperemia as measures of coronary flow reserve.* The American Journal Of Cardiology, 1974. **33**(1): p. 87-94.
37. Young, D.F., *Fluid mechanics of arterial stenoses.* Transactions of the ASME. Journal of Biomechanical Engineering, 1979. **101**(3): p. 157-75.
38. Bark Jr, D.L. and D.N. Ku, *Wall shear over high degree stenosis pertinent to atherothrombosis.* Journal of Biomechanics, **Accepted**.
39. Ma, P., X. Li, and D. Ku, *Heat and mass transfer in a separated flow region for high Prandtl and Schmidt numbers under pulsatile conditions.* International Journal of Heat and Mass Transfer, 1994. **37**(17): p. 2723-2736.
40. Young, D.F. and F.Y. Tsai, *Flow characteristics in models of arterial stenoses. I. Steady flow.* Journal of Biomechanics, 1973. **6**(4): p. 395-410.
41. Markou, C.P., et al. *Role of high wall shear rate on thrombus formation in stenoses*. in *Advances in Bioengineering*. 1993. New Orleans: ASME.
42. Sorensen, E.N., et al., *Computational Simulation of Platelet Deposition and Activation: II. Results for Poiseuille Flow over Collagen.* Annals of Biomedical Engineering, 1999. **27**(4-12): p. 449-458.
43. Bark, D. and D. Ku, *Wall shear over high degree stenoses pertinent to atherothrombosis.* Journal of Biomechanics, 2010.
44. Tang, D., et al., *Simulating cyclic artery compression using a 3D unsteady model with fluid-structure interactions.* Computers & Structures, 2002. **80**(20-21): p. 1651-1665.
45. Downing, J.M. and D.N. Ku, *Effects of frictional losses and pulsatile flow on the collapse of stenotic arteries.* Journal Of Biomechanical Engineering, 1997. **119**(3): p. 317-324.

CHAPTER 5. CONCLUSIONS AND FUTURE WORK

Discussion and Conclusions

This dissertation explored the hemodynamics and thrombosis of severe stenoses. Studies described in chapter 2 demonstrate that shear rates exceed values that have been historically considered as pertinent to thrombus growth. Maximum shear rates in severe short stenoses were found to exceed 250,000 s^{-1} (9,500 dynes/cm^2) and can reach a peak value of 425,000 s^{-1} for a 98% stenosis. Pulsatility and stenosis eccentricity were found to have minor effects on the maximum wall shear rates in severe stenoses. In contrast, increasing the stenosis length reduced the maximum shear to 107,000 s^{-1} (98% stenosis), while surface roughness could increase focal wall shear rates to a value reaching 610,000 s^{-1} (90% stenosis). Previous studies have shown that maximum shear can reach 84,000 s^{-1} in canine arteries with a moderate stenosis of 65% [1]. However, this shear rate increases as thrombus is superimposed over a stenosis, progressing the stenosis until a 98% stenosis by diameter, where flow becomes Stokes-like, based on our findings. Despite these findings, thrombus formation is commonly studied at shear rates below of 10,000 s^{-1} [2-11]. Some studies have looked at thrombus growth at shear rates reaching 32,000 s^{-1} [2-3, 12], but little is known for thrombus growth at larger shear rates. Similarly platelet adhesion studies and binding protein studies are studied in the same low to moderate shear range [8, 13-14].

We utilized high shear rates found in chapter 2 to analyze *in vitro* experiments of thrombus growth over collagen coated stenoses in chapter 3. Thrombus growth rates were evaluated for shear rates reaching 100,000 s^{-1}. We evaluated over 25,000

observation points (temporally and spatially independent) along a wide range of shear rates to quantify local thrombus thickness and growth rates. For thrombus thickness greater than 10 μm, we found a strong positive correlation between thrombus growth rates and shear rates up to 6,000 s^{-1} after a log-log transformation ($r=0.85$, $p<0.0001$). Thrombus growth rate values from the literature followed a similar trend with quantities below our experimental thrombus growth rates [2-3, 10]. Shear rates greater than 6,000 s^{-1} resulted in a small decreasing trend of growth rates relative to wall shear. Despite the decrease, growth rates at pathologic shear rates were typically 2-4 times greater than for physiologic shear rates below 400 s^{-1}. Prior to 10 μm of thrombus thickness, an initial lag time was found before thrombus could be detected. The lag time for initial deposition was inversely correlated with wall shear rate with a log-log transformation ($r = -0.85$, $p<0.0001$). Platelet activation lag time relative to shear rates follows a similar trend [15]. Activation may be occurring on platelets that have deposited with a static or transient bond since activation levels predicted from shear histories in chapter 2 and the literature [16] were below required threshold values. Thrombus growth in this study occurred in the throat of the stenosis where shear rates were highest. These results did agree with much of the literature for whole blood, but do not agree with all previous results that studied thrombus growth while using platelet rich plasma instead of whole blood [17-18]. These studies found most thrombus to form distal to the stenosis. Very little thrombus formed at the throat of the stenosis. These studies were performed at much higher upstream Re, reaching up to 3,600, and contained flow that separated near the peak of the stenosis with a downstream recirculation region. Flow in our chapter 3 study did not separate, until thrombus nearly occluded the test section. Locations of flow separation

and large decreasing microgradients have been reported to have thrombus deposition [19].

To further evaluate the mechanisms behind thrombus growth, to investigate the discrepancy in data with platelet rich plasma studies, and to develop a predictive model of thrombus, we utilized species transport simulations in chapter 4. Predicted thrombus growth rates were compared with the experimental findings of chapter 3 to determine if the growth rate is limited by platelet arrival wait at the wall or bonding kinetics for platelets already at the wall. Thrombus growth rates predicted from studies with an infinite binding rate were limited by the rate of platelet arrival and the results matched experimental platelet deposition rates for shears less than 6,000 s^{-1}. However, transport was much greater than platelet deposition rates for higher shear and increased relative to shear without bound. Therefore, we applied a kinetic rate constant of $1x10^{-4}$ m/s to control the binding rate of platelets at the surface. The rate constant is analogous to reaction rates at the wall between platelet glycoproteins and thrombus. Thrombus growth rates and thrombus volume growth with respect to time were predicted within a standard deviation of experimental results, when the kinetic binding rate was included in the model. These results indicate that thrombus growth becomes rate-limited by the kinetic binding of platelets to thrombus once shear rates exceed 6,000 s^{-1}. An extension of the model predicted thrombus occlusion times for varying stenosis severity. More severe stenoses result in faster occlusion times and the time course of deposition in the model matches that of experiments. When platelet arrival rates were based on thermal diffusivity and margination only, thrombus growth rates were very low relative to experiments of chapter 3. Thrombus also grew uniformly along the stenosis when the platelet arrival rate was dependent only on thermal diffusivity. Since the diffusion rate, associated with whole blood, decreases to thermal diffusivity as hematocrit decreases, the thermal diffusivity studies are representative of platelet rich plasma. The difference

between thermal diffusivity studies and enhanced diffusivity studies may be a possible explanation of the discrepancy between thrombus growth rates found with platelet rich plasma [17-18] and thrombus growth rates found in chapter 3.

For regions of high shear, thrombus growth rates are predicted to be rate-limited by kinetic bonding between platelets and thrombus based on our results from chapter 3 and 4. Platelets passing a point at rapid speeds near the apex of the stenosis are calculated to be captured onto the mural thrombus within 5-10 μs based on the high shear results of chapter 2 and 3, thus having to form bonds in a very short amount of time. Once bound, platelets would be required to resist a drag force of 17,500 pN. The hemodynamic drag force applied to a platelet attached to the surface is more than 2 orders of magnitude greater than the reported optimal bond strength (75 pN for longest bond lifetime [20]) between platelet GPIbα and the vWF A1 domain.

This study provides an analysis of thrombosis at high shear rates that establishes the strong quantitative relationship between hemodynamics and platelet deposition. Thrombosis growth continues at exceedingly high shear rates and the accumulation of platelets can be predicted from calculations of the patient-specific local hemodynamics. Lastly, the analysis implies that binding kinetics is secondary to hemodynamics for shears less than 6000 s^{-1}. Alternatively, changes in local hemodynamics could alter the development of occlusive thrombosis independent of receptor-ligand kinetics.

Limitations

The studies found in chapters 2-4 have several limitations. All of the studies only considered rigid walls typical of a calcific atheroma. However hemodynamics may be affected in the presence of a non-rigid stenosis that could result in potential vessel collapse from low Bernoulli pressures near the stenosis apex [21-22]. The flow rate and geometry of chapter 3 were not varied and therefore the results may not extend to different boundary conditions. However, the results are consistent with studies that have

used different flow rates or geometry. Thrombus in the study is assumed to act as a solid impenetrable substance, whereas physical thrombus may have porous characteristics. In chapter 3, we did not standardize results based on platelet concentrations, hematocrit, and platelet volume, even though these parameters may affect thrombus growth. Instead, the parameters were averaged within the data. Activation states and aggregation of platelets may occur while transporting the blood used for the experiments of chapter 3. The verification of the model found from chapter 4 was only performed relative to thrombus growth found in chapter 3, which were consistent with previously reported results. Platelet and RBC distributions of chapter 4 were only verified for straight sections of flow and distributions over the stenosis were not verified with experimental results.

Future Work

Thrombus studies considering thrombus growth at high shear, exceeding values of 32,000 s^{-1}, may result in a new understanding of the binding mechanisms, since bonds must form extremely fast and must be strong enough to resist the high drag forces over platelets binding at high shear. For very high shear, we show the thrombus growth is predicted to be rate-limited by kinetics. Therefore, alterations to the binding mechanisms would impact the rate of thrombus growth and possibly the likelihood of thrombus occlusion. The model developed in the current work could be extended by including and blocking specific binding proteins to determine what binding proteins may limit the rate of thrombus growth at high shear. Those proteins could be targeted by therapies in attempt to reduce the risk of thrombus occlusion. Therapies that can target high shear mechanisms, without affecting low shear mechanisms may be able to selectively target occluding thrombus on an atheroma without affecting normal hemostasis functions.

Results from the present work shows a transport-limited process of thrombus growth for wall shear rates below 6,000 s^{-1}. Therefore, hematocrit levels or geometrical modifications could alter transport-limited growth at this range of shear to determine if

thrombus growth rates can be increased or decreased. Decreasing the rate of thrombus growth could prevent thrombus occlusion. Such studies could be used to develop medical devices or therapies that target transport restrictions to minimize the risk of thrombosis by minimizing local shear rates. It may be possible to reduce the requirement for anticoagulants or anti-platelet drugs when implanting cardiovascular medical devices.

References

1. Strony, J., et al., *Analysis of shear stress and hemodynamic factors in a model of coronary artery stenosis and thrombosis.* American journal of physiology. Heart and circulatory physiology, 1993. **265**(5): p. H1787-1796.
2. Barstad, R.M., et al., *A perfusion chamber developed to investigate thrombus formation and shear profiles in flowing native human blood at the apex of well-defined stenoses.* Arteriosclerosis And Thrombosis: A Journal Of Vascular Biology / American Heart Association, 1994. **14**(12): p. 1984-1991.
3. Barstad, R.M., P. Kierulf, and K.S. Sakariassen, *Collagen induced thrombus formation at the apex of eccentric stenoses--a time course study with non-anticoagulated human blood.* Thrombosis And Haemostasis, 1996. **75**(4): p. 685-692.
4. Hubbell, J.A. and L.V. McIntire, *Platelet active concentration profiles near growing thrombi. A mathematical consideration.* Biophysical journal., 1986. **50**(5): p. 937-945.
5. Alevriadou, B.R., et al., *Real-time analysis of shear-dependent thrombus formation and its blockade by inhibitors of von Willebrand factor binding to platelets.* Blood., 1993. **81**(5): p. 1263-1276.
6. Sakariassen, K.S., et al., *Collagen type III induced ex vivo thrombogenesis in humans. Role of platelets and leukocytes in deposition of fibrin.* Arteriosclerosis, Thrombosis, And Vascular Biology, 1990. **10**(2): p. 276-284.
7. Turitto, V.T. and H.R. Baumgartner, *Platelet deposition on subendothelium exposed to flowing blood: mathematical analysis of physical parameters.* Transactions - American Society for Artificial Internal Organs., 1975. **21**: p. 593-601.
8. Savage, B., E. Saldívar, and Z.M. Ruggeri, *Initiation of platelet adhesion by arrest onto fibrinogen or translocation on von Willebrand factor.* Cell., 1996. **84**(2): p. 289-297.
9. Markou, C.P., Lindahl, A.K., Siegel, J.M., Ku, D.N., Hanson, S.R., *Effects of blocking the platelet GPIb interaction with von Willebrand Factor under a range of shearing forces.* Annals Of Biomedical Engineering, 1993. **21**: p. 220.
10. Markou, C.P., et al. *Role of high wall shear rate on thrombus formation in stenoses*. in *Advances in Bioengineering*. 1993. New Orleans: ASME.
11. Badimon, L., et al., *Influence of arterial damage and wall shear rate on platelet deposition. Ex vivo study in a swine model.* Arteriosclerosis (Dallas, Tex.), 1986. **6**(3): p. 312-320.
12. Ku, D.N. and C.J. Flannery, *Development of a flow-through system to create occluding thrombus.* Biorheology, 2007. **44**(4): p. 273-284.
13. Ruggeri, Z.M., et al., *Activation-independent platelet adhesion and aggregation under elevated shear stress.* Blood, 2006. **108**(6): p. 1903-1910.

14. Schneider, S.W., et al., *Shear-induced unfolding triggers adhesion of von Willebrand factor fibers.* Proceedings Of The National Academy Of Sciences Of The United States Of America, 2007. **104**(19): p. 7899-7903.
15. Hellums, J.D., *1993 Whitaker Lecture: biorheology in thrombosis research.* Annals Of Biomedical Engineering, 1994. **22**(5): p. 445-455.
16. Wootton, D.M., *Mechanistic Modeling of Occlusive Arterial Thrombosis*, in *Mechanical Engineering*. 1998, Georgia Institute of Technology: Atlanta. p. 421.
17. Schoephoerster, R.T., et al., *Effects of local geometry and fluid dynamics on regional platelet deposition on artificial surfaces.* Arteriosclerosis and thrombosis : a journal of vascular biology., 1993. **13**(12): p. 1806-1813.
18. Bluestein, D., et al., *Fluid mechanics of arterial stenosis: relationship to the development of mural thrombus.* Annals Of Biomedical Engineering, 1997. **25**(2): p. 344-356.
19. Nesbitt, W.S., et al., *A shear gradient-dependent platelet aggregation mechanism drives thrombus formation.* Nat Med, 2009. **15**(6): p. 665-673.
20. Yago, T., et al., *Platelet glycoprotein Ibalpha forms catch bonds with human WT vWF but not with type 2B von Willebrand disease vWF.* The Journal Of Clinical Investigation, 2008. **118**(9): p. 3195-3207.
21. Tang, D., et al., *Simulating cyclic artery compression using a 3D unsteady model with fluid-structure interactions.* Computers & Structures, 2002. **80**(20-21): p. 1651-1665.
22. Downing, J.M. and D.N. Ku, *Effects of frictional losses and pulsatile flow on the collapse of stenotic arteries.* Journal Of Biomechanical Engineering, 1997. **119**(3): p. 317-324.

APPENDICES

Appendix A: Determining Viscous Loss over Stenosis (Chapter 2)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Integrate shear over stenosis
%David Bark
%11/11/10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all
d=sym('d');
x=sym('x');
x0=sym('x0');
st=sym('st');
d=3/1000;
x0=2/1000;
stt=[0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 92 95 97 98 99];
sr=[];
rr=[];
L=50*d;
% Choose a stenosis % by diameter to evalute
i=1
st=stt(i);
r =  d*(.5-(st/800)*(1-cos(pi*(x/x0+1))).^2);
r = @(x)(d*(.5-(99/800)*(1-cos(pi*(x/(2/1000)+1))).^2)).^(-4)
r = quad(r,-x0,x0);
r=8*.0038*r/(pi);
r=abs(r)+8*.0038*L/(pi*(((d/2)^4)))+1.46654605e9;
Q=13332/r;%pascal
sr=[sr 4*Q/pi/((100-st)*.5*d/100)^3];
figure
plot(stt,sr)
hold on
st=[50 75 85 90 93 95 97 98 99];
b=[50000 271979 349020 383293 401021 403195 393792 424772 308111];
plot(st,b,'o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidth',5,'Li
neStyle','-','Color','k')
xlim([50 100]);
% ylim([0 500000]);

set(gca,'FontSize',35);
pos = get(gca,'position');
  set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
xlabel('% Stenosis','FontSize',40);
```

```
ylabel('Shear Rate (1/s)','FontSize',40);
grid on
```

Appendix B: Automate Grid Generation (Chapter 2)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Analyze images to get thrombus deposition
%David Bark
%11/11/10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all
clear java
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A=0;%.3;%.1;%.1;
pp=2*6666; % 6666 if pres, 2*6666 if scale
% zf=5.25;
zf=4;
s0=.4;
r0=1.5;
tri=1;
p=pp;
if (p~=6666)
   p=13332;
   if s0<.75
      p=p-(-(1.93e-4)*(100*s0).^2+(9.64e-3).*100.*s0+8.5e-1).*p;
   else
      p=p-(.5+(s0-.75)*(-.5)/(1-.75)).*p;%get your lazy ass off...your lazy ass
   end
end
p=pp;
z=[0:.005:1].*zf;
z=[0:.02:1].*zf;
% s=r0.*s0.*.25.*((1-cos(2.*pi.*z./zf)).^2);%+(A-A.*abs(z-.5.*zf)/(.5.*zf)).*cos(abs(z-
.5.*zf)/.025);

z=(z-.5.*zf)/(.5*zf);
s=(s0.*.25.*((1-cos(pi.*(z+1))).^2));
% z=[-2 z 2];
% s=[1 s 1];
% figure
% plot(z,s,'k')
% set(gca,'FontSize',25);
% axis equal
% xlim([-2 2]);
% ylim([-2 2]);
% xlabel('x*','FontSize',30);
% ylabel('r*','FontSize',30);
```

```
% hold on
% plot(z,-s,'k')
plot(z,r0-r0*s,'r')
hold on
% x=[0 1/10 3/10 4/10 5/10 6/10 6.2/10 9/10 1]*zf;
% y=[0 0 A/8 A/3 A/8 A A/8 0 0];
% x=[3 3.5 4 4.5 5 5.5 6 6.1 6.2]*zf/10;
% y=[0 A/6 A/3 .5*(A/3+A/8) A/8 .5*(A/8+A) A A/2 0];
% xxx=.2.*((1/(.6*zf))*z.*(z<=.6*zf*ones(1,length(z)))+(1-(1/(.1*zf))*(z-
.6*zf)).*(z>.6*zf*ones(1,length(z))).*(z<.7*zf*ones(1,length(z)))).*(sin(4*pi*(z-
.13*zf)/zf)+1);
T=.7
om=.5/T;
xxx=.2.*((sin(2*pi*(om)*(z))).^2).*(abs(z)<T);
% xxx=.2.*(abs(z)<T).*(((sin(2*pi*(om)*(z))).^2)-(abs(z)>T/2)*;

T=.2
x=[-1 -.5 -.37 0 .37 .5 1];
y=[0 0 T 0 T 0 0];
xxx=csaps(x,y,1,z);
xxx=xxx.*(xxx>0)*(A>0);

% xxx=0;
plot(z,xxx,'g')

% plot(z,r0-s,'r');

% y=csaps(x,y,1,z);
% % s=s+y;
% % plot(z,g)
% % hold on
% % plot(x,y,'r')
x=[0:.01:1];
xx=A.*(spatialPattern([1,length(z)],-2)/zf);
xx(1)=0;
xx(length(xx))=0;
plot(z,xxx+r0*s,'m')
xxx=xxx+xx;
xxx=0;
s=xxx+r0*s;%.*(xxx>0)+s;



r=(r0-s);%
plot(z,r)
axis equal
```

```
z=.5*zf*z;
x=z;
% ylim([0 4]);
% figure
% plot(z,xx)
figure
plot(z,xx)
% %
% %
le=length(r);
% x=z-.5.*zf;
% % t=sym('t');
% % diff(r0.*s0.*.25.*((1-cos(2.*pi.*t./zf)).^2)-(A-A.*abs(t-.5.*zf)/(.5.*zf)).*cos(abs(t-
.5.*zf)/.025))
% %
% % arl=(int(double(sqrt((),t,0,zf)));
% % arl=sqrt((r0.*s0.*.25.*diff(((1-cos(2.*pi.*t./zf)).^2),t)).^2+1)
% % arl=arl*750;

F = @(t)sqrt((297/800*(1-cos(1/2*pi*t))*sin(1/2*pi*t)*pi).^2+1);
arl = quadv(F,0,zf);
% arl=180*arl;
arl=80*arl;
pood=ceil(arl);
if(rem(pood,2))
pood=floor(arl);
end
pood
pood2=125*zf;
% if(rem(pood2,2))
% pood2=floor(arl);
% end
pood2

if ((tri==0)&&(A==0)&&(zf==4)&&(pp==6666)&&(tri==0))
   y='G:\\shr_stdy2\\shear_exp\\4mm\\';
elseif ((tri==0)&&(A==0)&&(zf==16)&&(pp==6666))
   y='G:\\shr_stdy2\\shear_exp\\16mm\\';
elseif ((tri==0)&&(A==0)&&(zf==4)&&(pp~=6666))
   y='G:\\shr_stdy2\\shear_exp\\scale4mm\\';
elseif ((tri==0)&&(A==0)&&(zf==16)&&(pp~=6666))
   y='G:\\shr_stdy2\\shear_exp\\scale16mm\\';
elseif ((tri==1)&&(A>0)&&(zf==16)&&(pp==6666))
   y='G:\\shr_stdy2\\shear_exp\\sinusoid_4mm\\';
elseif ((tri==1)&&(A>0)&&(zf==16)&&(pp~=6666))
   y='G:\\shr_stdy2\\shear_exp\\sinusoid_scale4mm\\';
```

```
elseif ((tri==1)&&(A==0)&&(zf==4)&&(pp==6666))
    y='G:\\shr_stdy2\\shear_exp\\4mm_tri\\';
elseif ((tri==1)&&(A==0)&&(zf==16)&&(pp==6666))
    y='G:\\shr_stdy2\\shear_exp\\16mm_tri\\';
elseif ((tri==1)&&(A==0)&&(zf==4)&&(pp~=6666))
    y='G:\\shr_stdy2\\shear_exp\\scale4mm_tri\\';
elseif ((tri==1)&&(A==0)&&(zf==16)&&(pp~=6666))
    y='G:\\shr_stdy2\\shear_exp\\scale16mm_tri\\';
elseif ((tri==1)&&(A>0)&&(zf==4)&&(pp==6666))
    y='G:\\shr_stdy2\\shear_exp\\sinusoid_4mm_tri\\';
elseif ((tri==1)&&(A>0)&&(zf==4)&&(pp~=6666))
    y='G:\\shr_stdy2\\shear_exp\\sinusoid_scale4mm_tri\\';
elseif ((tri==1)&&(zf==2))
    y='G:\\shr_stdy2\\shear_exp\\scale2mm_tri\\';
end

% y='G:\\shr_stdy2\\shear_exp\\';
y='G:\\shr_stdy2\\shear_exp\\';
y


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fid = fopen('G:\shr_stdy2\sten.jou', 'wt');


for(i=1:le)
    fprintf(fid, 'vertex create "r%i" coordinates %6.3f%6.3f%6.3f\n', i, x(i), r(i), 0.0);
end

fprintf(fid, 'edge create "sten" nurbs ');
for(i=1:le)
    fprintf(fid, '"r%i" ', i);
end
fprintf(fid, 'interpolate\n');
fprintf(fid, 'vertex create "x_prox_wall" coordinates %6.3f%6.3f%6.3f\n', -20*r0, r0,
0.0);
fprintf(fid, 'vertex create "x_prox_ax" coordinates %6.3f%6.3f%6.3f\n', -20*r0, 0.0, 0.0);
fprintf(fid, 'vertex create "x_dist_wall" coordinates %6.3f%6.3f%6.3f\n', 20*r0, r0, 0.0);
fprintf(fid, 'vertex create "x_dist_ax" coordinates %6.3f%6.3f%6.3f\n', 20*r0, 0.0, 0.0);

fprintf(fid, 'vertex create "x_dist_wall3" coordinates %6.3f%6.3f%6.3f\n', 60*r0, r0, 0.0);
fprintf(fid, 'vertex create "x_dist_ax3" coordinates %6.3f%6.3f%6.3f\n', 60*r0, 0.0, 0.0);

fprintf(fid, 'vertex create "x_dist_wall2" coordinates %6.3f%6.3f%6.3f\n', 58*r0, r0, 0.0);
fprintf(fid, 'vertex create "x_dist_ax2" coordinates %6.3f%6.3f%6.3f\n', 58*r0, 0.0, 0.0);
```

```
fprintf(fid, 'vertex create "x_prox_ax_mid" coordinates %6.3f%6.3f%6.3f\n', x(1), 0.0,
0.0);
fprintf(fid, 'vertex create "x_dist_ax_mid" coordinates %6.3f%6.3f%6.3f\n', x(le), 0.0,
0.0);

fprintf(fid, 'edge create "wall_prox" "in" "ax_prox" "ax_mid" "ax_dist" "out" "wall_dist"
straight "r%i" "x_prox_wall" "x_prox_ax" "x_prox_ax_mid" "x_dist_ax_mid"
"x_dist_ax" "x_dist_wall"  "r%i"\n', 1,le);
fprintf(fid, 'edge create "mid_prox" straight "r%i" "x_prox_ax_mid"\n', 1);
fprintf(fid, 'edge create "mid_dist" straight "r%i" "x_dist_ax_mid"\n', le);

fprintf(fid, 'edge create "por_wall" "por_vert" "por_ax" straight "x_dist_wall2"
"x_dist_wall3" "x_dist_ax3" "x_dist_ax2"\n');

fprintf(fid, 'face create "mid" wireframe "sten" "mid_prox" "ax_mid" "mid_dist" real\n');
fprintf(fid, 'face create "prox_mid" wireframe "wall_prox" "in" "ax_prox" "mid_prox"
real\n');
fprintf(fid, 'face create "dist_mid" wireframe "ax_dist" "out" "wall_dist" "mid_dist"
real\n');
fprintf(fid, 'edge create "ax_dist2" "out2" "wall_dist2" straight "x_dist_ax_mid"
"x_dist_ax2" "x_dist_wall2" "x_dist_wall"\n');
fprintf(fid, 'face create "outer" wireframe "out" "ax_dist2" "out2" "wall_dist2" real\n');

fprintf(fid, 'face create "por" wireframe "por_wall" "por_vert" "por_ax" "out2" real\n');

% fprintf(fid, 'edge mesh "sten" ratio1 %f ratio2 %f size .008\n',1/1.005, 1/1.005);
% fprintf(fid, 'edge mesh "sten" firstlength ratio .02 intervals %f\n',pood);
fprintf(fid, 'edge mesh "wall_prox" successive ratio1 1.005 size 0.06\n');
fprintf(fid, 'edge mesh "ax_prox" successive ratio1 %f size 0.06\n',1/1.005);
% fprintf(fid, 'edge mesh "wall_prox" successive size 0.02\n');
% fprintf(fid, 'edge mesh "ax_prox" successive size 0.02\n',1/1.005);

% fprintf(fid, 'edge split "sten" percentarclength 0.5 connected\n');

fprintf(fid, 'edge mesh "wall_dist" successive ratio1 %f size 0.06\n',1/1.005);

fprintf(fid, 'edge mesh "sten" firstlength ratio1 0.02 ratio2 0.02 size .01\n');
fprintf(fid, 'edge mesh "ax_mid" firstlength ratio1 0.02 ratio2 0.02 size .01\n');
% fprintf(fid, 'edge mesh "wall_prox" successive ratio1 1.005 size 0.035\n');
% fprintf(fid, 'edge mesh "wall_dist" successive ratio1 %f size 0.035\n',1/1.005);

fprintf(fid, 'edge mesh "ax_dist" successive ratio1 1.005 size 0.06\n')
%
fprintf(fid, 'edge mesh "wall_dist2" successive size 0.1\n');
fprintf(fid, 'edge mesh "ax_dist2" successive size 0.1\n');
```

```
fprintf(fid, 'edge mesh "por_wall" successive size 0.1\n');
fprintf(fid, 'edge mesh "por_ax" successive size 0.1\n');
fprintf(fid, 'edge mesh "por_vert" successive ratio1 %f size 0.012\n',1.03);%ratio1 1.112

fprintf(fid, 'edge mesh "mid_prox" successive ratio1 1.03 size 0.012\n');%ratio1 1.112
ratio1 1.03 size 0.019
fprintf(fid, 'edge mesh "mid_dist" successive ratio1 1.03 size 0.012\n');%ratio1 1.112
fprintf(fid, 'edge mesh "in" successive ratio1 1.03 size 0.012\n');%ratio1 1.112
fprintf(fid, 'edge mesh "out" successive ratio1 %f size 0.012\n', 1/1.03);%ratio1 1.112
fprintf(fid, 'edge mesh "out2" successive ratio1 %f size 0.012\n',1/1.03);%ratio1 1.112
% fprintf(fid, 'edge mesh "ax_mid" ratio1 %f ratio2 %f size .008\n',1/1.005, 1/1.005);

% fprintf(fid, 'edge mesh "ax_mid" intervals %f\n',pood);
if ((A==0)&&(s0<.97))
   fprintf(fid, 'blayer create first 0.0005 growth 1.1 total 0.0474359 rows 15 transition 1
trows 0 uniform\n');
   fprintf(fid, 'blayer attach "b_layer.1" face "mid" edge "sten" add\n');
elseif ((A==0)&&(s0>.97))
   fprintf(fid, 'blayer create first 0.0003 growth 1.1 total 0.0474359 rows 15 transition 1
trows 0 uniform\n');
   fprintf(fid, 'blayer attach "b_layer.1" face "mid" edge "sten" add\n');
end
fprintf(fid, 'face mesh "por" map\n');
fprintf(fid, 'face mesh "outer" map\n');
fprintf(fid, 'face mesh "dist_mid" map\n');
fprintf(fid, 'face mesh "prox_mid" map\n');
if (tri==1)&&(s0>.6)
   fprintf(fid, 'face mesh "mid" triangle\n');
   fprintf(fid, 'edge mesh "ax_mid" ratio1 %f ratio2 %f intervals %f\n',1/1.01,
1/1.01,pood2);
elseif (s0>.6)
   fprintf(fid, 'face mesh "mid" tripave\n');
   fprintf(fid, 'edge mesh "ax_mid" ratio1 %f ratio2 %f intervals %f\n',1/1.01,
1/1.01,pood2);
else
   fprintf(fid, 'face mesh "mid" map\n');
end
% fprintf(fid, 'face mesh "mid" map size 1\n');
% fprintf(fid, 'face smooth "outer" "mid" free centroidarea\n');
% fprintf(fid, 'face smooth "mid" free lwlaplacian\n');
% if (s0>.975)
%     fprintf(fid, 'blayer create first 0.00025 growth 1.1 total 0.0474359 rows 15 transition
1 trows 0 uniform\n');
%     fprintf(fid, 'blayer attach "b_layer.1" face "mid" "mid" "mid" "outer" edge "sten"
"wall_prox" "wall_dist" "wall_dist2" add\n');
```

```
%     fprintf(fid, 'edge mesh "in" successive ratio1 1.02 size 0.03\n');%ratio1 1.112
%     fprintf(fid, 'edge mesh "ax_mid" ratio1 %f ratio2 %f size .008\n',1/1.005, 1/1.005);
%     fprintf(fid, 'edge mesh "ax_prox" successive ratio1 %f size 0.06\n',1/1.005);
%     fprintf(fid, 'edge mesh "ax_dist" successive ratio1 1.005 size 0.06\n');
%     fprintf(fid, 'edge mesh "out"  size 0.1\n');%successive ratio1 %f,1/1.112
%     fprintf(fid, 'face mesh "mid" tripave\n');
%     fprintf(fid, 'face mesh "outer" map\n');
% else
%     fprintf(fid, 'edge mesh "ax_dist" successive ratio1 1.005 size 0.045\n');
%     fprintf(fid, 'edge mesh "ax_prox" successive ratio1 %f size 0.045\n',1/1.005);
%     fprintf(fid, 'edge mesh "in" successive ratio1 1.02 size 0.03\n');%ratio1 1.112
%     fprintf(fid, 'face mesh "mid" "outer" map\n');

% end
fprintf(fid, 'physics create "pin" btype "PRESSURE_INLET" edge "in"\n');
fprintf(fid, 'physics create "porj" btype "POROUS_JUMP" edge "out2"\n');
fprintf(fid, 'physics create "pout" btype "PRESSURE_OUTLET" edge "por_vert"\n');
fprintf(fid, 'physics create "wall" btype "WALL" edge "por_wall" "wall_prox" "sten"
"wall_dist" "wall_dist2"\n');
fprintf(fid, 'physics create "ax" btype "AXIS" edge "por_ax" "ax_prox" "ax_mid"
"ax_dist" "ax_dist2"\n');
fprintf(fid, 'solver select "FLUENT 5/6"\n');
fprintf(fid, ['export fluent5 "' y 'sten%i.msh" nozval'],s0.*100);

fclose(fid);
if 1==0
   system('C:\Fluent.Inc\ntbin\ntx86\gambit.exe -r2.4.6 -inp G:\shr_stdy2\sten.jou -new')

   %
   fid = fopen('G:\shr_stdy2\stenfl.jou', 'wt');
   fprintf(fid, ['file read-case ' y 'sten%i.msh\n'],s0.*100);
   fprintf(fid, 'grid scale .001 .001\n');
   fprintf(fid, 'define models axisymmetric yes\n');
   fprintf(fid, 'define materials change-create air plasma y constant 1025 n n y constant
.0038 n n n n n n n \n');
   fprintf(fid, 'define boundary-conditions fluid fluid y plasma n n y n n \n');
   fprintf(fid, 'define boundary-conditions pressure-inlet pin n %f n 0 n y \n',p);%pascal
   fprintf(fid, 'solve initialize initialize-flow \n');
   fprintf(fid, 'solve monitors residual plot n \n');
   fprintf(fid, 'solve monitors residual print y \n');
   fprintf(fid, 'solve monitors residual convergence-criteria .0000001 .001 .001\n');
   % fprintf(fid, 'adapt refine-bndry-cells 4 () 10 0 1000000 y\n');
   % fprintf(fid, 'adapt refine-bndry-cells 4 () %f 0 1000000 y\n',30);
   fprintf(fid, 'file auto-save case-frequency 500\n');
   fprintf(fid, 'file auto-save data-frequency 500\n');
   fprintf(fid, 'file auto-save overwrite-existing-files y\n');
```

```
    fprintf(fid, 'file auto-save max-files 1\n');
    fprintf(fid, 'define boundary-conditions porous-jump porj %f 1 0
\n',.00000036291);%pascal
    fprintf(fid, 'solve iterate 1000000\n');
    %
    fclose(fid);
    system('C:\Fluent.Inc\ntbin\win64\fluent.exe -r6.3.26 2ddp -i G:\shr_stdy2\stenfl.jou')
end
% % % %
% y='G:\\shr_stdy2\\8mm100mmhg\\';
% xx=[-20.*r0:.01:2.*20.*r0];
% shrinf=[];
% for j=[20 30 40 60 70 80 90];
%     fid = fopen([y,num2str(j),'_shrmag'],'r');
%     shrdat = textscan(fid,'%f %f',-1, 'headerlines', 4);
%     shrdat=cell2mat(shrdat);
%     shrx=1000*(shrdat(:,1));
%     shrm=(shrdat(:,2));
%     fclose(fid);
%      shrinf=[shrinf csaps(shrx,shrm,.5,xx)'];
% end
% contourf([20 30 40 60 70 80 90],xx,shrinf,30);
% ylim([-2 2]);
```

Appendix C: Plotting Figures (Chapter 2)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plotting figures for chapter 2
%David Bark
%11/11/10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all
format short
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
%Evaluate Shear
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% k=[162: 4: 294];
% t=[0:.04:(max(k)-min(k))*.01];
% Q=(60000000)*[2.07E-08    1.66E-08    1.19E-08    2.10E-08    3.37E-08    3.78E-08
3.64E-08    3.56E-08    3.34E-08    3.20E-08    3.04E-08    2.86E-08    2.73E-08    2.56E-
08    2.35E-08    2.30E-08    1.95E-08    8.60E-09    2.21E-09    1.10E-08    2.07E-08
1.67E-08    1.19E-08    2.09E-08    3.37E-08    3.78E-08    3.64E-08    3.49E-08    3.34E-
08    3.20E-08    3.04E-08    2.87E-08    2.73E-08    2.56E-08];

k=[310:2:316 320:2:334 338 342:2:370 374:2:388];
Q=[2.4945e-8,1.078e-8,6.32e-9,1.429e-8,4.90877e-8,5.92667e-8,5.793e-8,4.8e-
8,3.7368e-8,3.4158e-8,4.225e-8,5.954376e-8,9.644334e-8,1.086e-7,1.0715e-
7,1.044736e-7,1.020933e-7,1.0013e-7,9.816781e-8,9.602693e-8,9.3867e-8,9.18167e-
8,8.9727e-8,8.735875e-8,8.4747e-8,8.2269e-8,8.025e-8,7.850797e-8,7.3432e-8,7e-
8,6.735824e-8,6.627843e-8,6.59882e-8,6.372563e-8,5.624757e-8,4.243648e-8];
k=[600:2:680];
Q=[8.9827e-8,8.74738e-8,8.486668e-8,8.237316e-8,8.033271e-8,7.858998e-8,7.6505e-
8,7.35835e-8,7.01782e-8,6.7447e-8,6.629631e-8,6.601807e-8,6.392134e-8,5.673111e-
8,4.317841e-8,2.597918e-8,1.1419e-8,6.235654e-9,1.348704e-8,3.012441e-8,4.815181e-
8,5.897024e-8,5.832122e-8,4.872431e-8,3.787109e-8,3.404034e-8,4.145076e-
8,5.833612e-8,7.860231e-8,9.56191e-8,1.055855e-7,1.085931e-7,1.072994e-
7,1.046351e-7,1.022226e-7,1.002456e-7,9.82925e-8,9.61599e-8,9.399515e-8,9.193968e-
8,8.986056e-8];

k=[600:2:679];
Q=[8.9827e-8,8.74738e-8,8.486668e-8,8.237316e-8,8.033271e-8,7.858998e-8,7.6505e-
8,7.35835e-8,7.01782e-8,6.7447e-8,6.629631e-8,6.601807e-8,6.392134e-8,5.673111e-
8,4.317841e-8,2.597918e-8,1.1419e-8,6.235654e-9,1.348704e-8,3.012441e-8,4.815181e-
8,5.897024e-8,5.832122e-8,4.872431e-8,3.787109e-8,3.404034e-8,4.145076e-
8,5.833612e-8,7.860231e-8,9.56191e-8,1.055855e-7,1.085931e-7,1.072994e-
```

```
7,1.046351e-7,1.022226e-7,1.002456e-7,9.82925e-8,9.61599e-8,9.399515e-8,9.193968e-
8];

k=[90:2:170];
Q=[3.8517E-08 3.47435E-08 2.75E-08 2.22E-08 2.33E-08 3.15E-08 4.40E-08 5.64E-08
6.51E-08 6.91E-08 6.94E-08 6.79E-08 6.63E-08 6.49E-08 6.37E-08 6.23E-08 6.09E-08
5.96E-08 5.83E-08 5.69E-08 5.53E-08 5.36E-08 5.21E-08 5.10E-08 4.98E-08 4.83E-08
4.61E-08 4.40E-08 4.28E-08 4.24E-08 4.20E-08 3.92E-08 3.26E-08 2.24E-08 1.16E-08
4.74E-09 5.30E-09 1.34E-08 2.52E-08 3.51E-08 3.85E-08];

t=.01*k;
t=t-min(t);
[a b]=min(Q);
k=[k([b:length(Q)]) k([1:(b-1)])];
Q=[Q([b:length(Q)]) Q([1:(b-1)])];


shrinf=[];
for j=k
%     fid = fopen(['G:\shr_stdy2\shear_exp\4mm_unsteady_double\sten95_double-
0',num2str(j)],'r');
if j<100
fid = fopen(['E:\shr_stdy2\shear_exp\unsteady_max\sten95-00',num2str(j)],'r');
else
   fid = fopen(['E:\shr_stdy2\shear_exp\unsteady_max\sten95-0',num2str(j)],'r');
end
   shrdat = textscan(fid,'%f %f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
   shrdat=cell2mat(shrdat);
   shrx=1000*(shrdat(:,2));
   shrm=(shrdat(:,4));
   shrm=-shrm.*sign((shrdat(:,5)));
   fclose(fid);
%     shrinf=[shrinf csaps(shrx,shrm,.5,ll.*x/1000)];
   shrinf=[shrinf shrm];
end
%
% x=[0:.000001:(.05*.0015)]';
% vel=[];
% fid = fopen(['G:\shr_stdy2\shear_exp\4mm_unsteady_double\sten95_double-
max_vwav_puls'],'r');
% shrdat = textscan(fid,'%f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
% shrdat=cell2mat(shrdat);
% vx=(shrdat(:,3));
% fclose(fid);
% figure
% plot(vx,shrdat(:,4),'k','LineWidth',6,'LineStyle','-')
```

```
% hold on
%
% fid = fopen(['G:\shr_stdy2\shear_exp\4mm_unsteady_double\sten95_double-
max_vwav_stead'],'r');
% shrdat = textscan(fid,'%f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
% shrdat=cell2mat(shrdat);
% vx=(shrdat(:,3));
% fclose(fid);
% plot(vx,shrdat(:,4),'k','LineWidth',6,'LineStyle','-.')
% % vel=csaps(vx,(shrdat(:,4)),1e-10,x);
% % figure
% % plot(x,vel,'k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot shear contour
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
figure
[C,h]=contourf(t/max(t),shrx/3,shrinf,20);

set(gca,'FontSize',35);
  set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% colormap((gray));


xlim([0 1]);
ylim([-.3 .3]);
xlabel('t/t_m_a_x','FontSize',40);
ylabel('x/r_0','FontSize',40);
title('Pulsatile Shear Rate','FontSize',40);
colorbar;
set(gca,'FontSize',35);
grid on;





figure
plot(t/max(t),max(shrinf),'k','LineWidth',4)
n=get(gca,'Ytick');
set(gca,'yticklabel',sprintf('%g |',n'));
set(gca,'FontSize',35);
pos = get(gca,'position');
  set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
xlim([0 1]);
```

```
xlabel('t/t_m_a_x','FontSize',40);
ylabel('Shear Rate (1/s)','FontSize',40);
grid on
hold on
plot([0 1],[403200 403200],'k','LineWidth',4,'LineStyle','--')
% plot([min(t) max(t)],[403342 403342],'k','LineWidth',4,'LineStyle','-.')
% plot([min(t) max(t)],[757848 757848],'k','LineWidth',4,'LineStyle','-.')
legend('Pulsatile','Steady')

figure
plot(t/max(t),Q/(6.86e-8),'k','LineWidth',4)
set(gca,'FontSize',35);
pos = get(gca,'position');
  set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
xlim([0 1]);
ylim([0 1.2]);
xlabel('t/t_m_a_x','FontSize',40);
ylabel('(Flow Rate)/(Flow Rate_S_t_e_a_d_y)','FontSize',40);
grid on

% figure
% [AX,H1,H2]=plotyy(t,Q,t,max(shrinf))
% % set(gca,'FontSize',25);
% % set(get(AX(1)),'FontSize',25);
% % set(get(AX(2)),'FontSize',25);
% set(get(AX(1),'Ylabel'),'String','Flow Rate (ml/min)','FontSize',25)
% set(get(AX(2),'Ylabel'),'String','Max Shear Rate (1/s)','FontSize',25)
% set(H1,'LineStyle','--')
% % xlim([min(t) max(t)]);
% % set(H2,'LineStyle','o')
% xlabel('Time (s)','FontSize',20);
% grid on
% % ylabel('Time (s)','FontSize',20);
% title('Max Thrombus Growth w.r.t. Time','FontSize',20);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%
% k=[50 75 85 90 93 95 98 99];
% x=[-30:.01:90]';
% t=k;
% shrinf=[];
% for j=k
%     fid = fopen(['G:\shr_stdy2\shear_exp\scale_16mm_final\sten',num2str(j)],'r');
%     shrdat = textscan(fid,'%f %f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
%     shrdat=cell2mat(shrdat);
%     shrx=1000*(shrdat(:,2));
%     shrm=(shrdat(:,4));
```

```
%     shrm=-shrm.*sign((shrdat(:,5)));
%     fclose(fid);
% %     shrinf=[shrinf shrm];
%     shrinf=[shrinf csaps(shrx,shrm,.5,x)];
% end
% x=x/3;
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %Plot shear contour
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% numberOfColors = 8; % Choose the number of colors
% nc = numberOfColors-1; % This value will be used in the code below to set the
number of colors as desired
% figure
% % shrinf=shrinf./max(shrinf(:));
% contourf(t,x,shrinf,7);
% % colorbar;
% set(gca,'FontSize',35);
% set(gca,'position',[0.3    0.2000    0.6    0.7250]);
% colormap(flipud(gray));
% % xlim([0 100]);
% ylim([-1 1]);
% xlabel('% Stenosis','FontSize',35);
% ylabel('x/r_0','FontSize',35);
% title('Shear Rate (1/s)','FontSize',35);
% grid on;
%
% cb=colorbar;
% i = findobj(cb,'type','image'); % Find the image inside the colorbar object
% minVal = min(shrinf(:)); % Find the limits of the data to set colorbar axes accordingly
% maxVal = max(shrinf(:));
% set(i,'cdata',[0:64/nc:64]','YData',[minVal maxVal]); % Set the colorbar's CData to
cover the whole color range, but only with the desired number of intervals
% set(cb,'yLim',[minVal maxVal],'ytick',[minVal:(maxVal-minVal)/(nc):maxVal]'); %
Set the colorbar ticks to sit in the center of the color bins
%
set(cb,'YTickLabel',{'0','20,000','40,000','60,000','80,000','100,000','120,000','140,000'},'F
ontSize',35)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k=[50 75 85 90 93 95 97 98 99];
x=[-30:.01:90]';
t=k;
```

```
shrinf=[];
for j=k
   fid = fopen(['E:\shr_stdy2\shear_exp\scale_4mm_final\sten',num2str(j)],'r');
   shrdat = textscan(fid,'%f %f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
   shrdat=cell2mat(shrdat);
   shrx=1000*(shrdat(:,2));
   shrm=(shrdat(:,4));
   shrm=-shrm.*sign((shrdat(:,5)));
   fclose(fid);
%     shrinf=[shrinf shrm];
   shrinf=[shrinf csaps(shrx,shrm,1,x)];
end
x=x/1.5;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
%Plot shear contour
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%
numberOfColors = 6; % Choose the number of colors
nc = numberOfColors-1; % This value will be used in the code below to set the number
of colors as desired
figure
% shrinf=shrinf./max(shrinf(:));
contourf(t,x,shrinf,[0 10000 20000 100000 350000]);
set(gca,'FontSize',35);
set(gca,'position',[0.3    0.2000    0.6    0.7250]);
% colorbar;
colormap(flipud(gray));
% xlim([0 100]);
ylim([-.5 .5]);
xlabel('% Stenosis','FontSize',40);
ylabel('(Axial Position)/r_0','FontSize',40);
title('Shear Rate (1/s)','FontSize',40);
grid on;
caxis([-20000 425000])
%specifiy colormap scaling
H=CBARF(shrinf,[0 10000 20000 100000 350000],'vertical','nonlinear');
set(H,'YTickLabel',{'0','10,000','20,000','100,000','350,000'},'FontSize',35)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
%Calculate flow separation points w.r.t. time
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k=[50 75 85 90 93 95 97 98 99];
separation=[];
ret=[];
sepx=[];
sepxs=[];
for j=k
   fid = fopen(['E:\shr_stdy2\shear_exp\scale_4mm_final\sten',num2str(j)],'r');
   shrdat = textscan(fid,'%f %f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
   shrdat=cell2mat(shrdat);
   shrx=1000*(shrdat(:,2));
   shrm=(shrdat(:,4));
   shrm=-shrm.*sign((shrdat(:,5)));
   fclose(fid);
   for i=2:length(shrx)
      if(((shrm(i)<0)&&(shrm(i-1)>=0))&&(shrx(i)>0)&&(j~=50))%&&(x(j)>0)
         ret=[ret shrx(i)];
         sepx=[sepx j];
      elseif(((shrm(i)>0)&&(shrm(i-1)<=0))&&(shrx(i)>0)&&(j==50))%&&(x(j)>0)
         ret=[ret shrx(i)];
         sepx=[sepx j];
      elseif(((shrm(i)>0)&&(shrm(i-1)<=0))&&(shrx(i)>0)&&(j~=50))%&&(x(j)>0)
         separation=[separation shrx(i)];
         sepxs=[sepxs j];
      elseif(((shrm(i)<0)&&(shrm(i-1)>=0))&&(shrx(i)>0)&&(j==50))%&&(x(j)>0)
         separation=[separation shrx(i)];
         sepxs=[sepxs j];
      end
   end
end

figure
plot(sepx,ret/1.5,'o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWid
th',5,'LineStyle','-','Color','k')
xlim([75 100]);
set(gca,'FontSize',35);
set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
xlabel('% Stenosis','FontSize',40);
ylabel('x_r_e_a_t_t_a_c_h_m_e_n_t/r_0','FontSize',40);
grid on;

figure
plot(sepxs,separation,'o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'Lin
eWidth',5,'LineStyle','-','Color','k')
xlim([75 100]);
```

```
set(gca,'FontSize',35);
set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
xlabel('% Stenosis','FontSize',40);
ylabel('x_r_e_a_t_t_a_c_h_m_e_n_t/r_0','FontSize',40);
grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Analyze images to get thrombus deposition
%David Bark
%08/21/09
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A=0;%.3;%.1;%.1;
zf=4;
s0=.5;
r0=1.5;
tri=1;
z=[0:.005:1].*zf;%you should check your code before you give to anyone.
z=(z-.5.*zf)/(.5*zf);
s=(s0.*.25.*((1-cos(pi.*(z+1))).^2));
figure
plot([1-s -1+s],[z.*.5.*zf./r0 fliplr(z.*.5.*zf./r0)],'LineWidth',5,'LineStyle','-','Color','k')
plot([1-s],[z.*.5.*zf./r0],'LineWidth',5,'LineStyle','-','Color','k')
% xlim([75 100]);
set(gca,'FontSize',35);

xlabel('r/r_0','FontSize',40);
ylabel('(Axial Position)/r_0','FontSize',40);
grid on;
axis equal
ylim([-.5 .5]);
xlim([0 1]);
set(gca,'position',[0.3    0.2000    0.6    0.7250]);
set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% k=[50 75 90 95 98 99];
% separation=[];
% sepx=[];
% for j=k
%     fid = fopen(['G:\shr_stdy2\shear_exp\scale_16mm_final\sten',num2str(j)],'r');
%     shrdat = textscan(fid,'%f %f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
%     shrdat=cell2mat(shrdat);
%     shrx=1000*(shrdat(:,2));
```

```
%     shrm=(shrdat(:,4));
%     shrm=-shrm.*sign((shrdat(:,5)));
%     fclose(fid);
%     for i=2:length(shrx)
%         if(((shrm(i)<0)&&(shrm(i-1)>=0))&&(shrx(i)>0))%&&(x(j)>0)
%             separation=[separation shrx(i)];
%             sepx=[sepx j];
%         end
%     end
% end
%
% figure
% plot(sepx,separation,'.')
% xlabel('Time (s)','FontSize',20);
% ylabel('Axial Location (mm)','FontSize',20);
% title('Flow Separation Point Along Stenosis','FontSize',20);
% grid on;

% %%%%dp contours%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% k=[90 95 98 99];
% x=[-30:.01:90]';
% t=k;
% shrinf=[];
% for j=k
%     fid = fopen(['G:\shr_stdy2\shear_exp\4mm_final\sten',num2str(j)],'r');
%     shrdat = textscan(fid,'%f %f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
%     shrdat=cell2mat(shrdat);
%     shrx=1000*(shrdat(:,2));
%     shrm=(shrdat(:,4));
%     shrm=-shrm.*sign((shrdat(:,5)));
%     fclose(fid);
% %     shrinf=[shrinf shrm];
%     shrinf=[shrinf csaps(shrx,shrm,.5,x)];
% end
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %Plot shear contour
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% figure
% shrinf=shrinf./max(shrinf(:));
```

```
% x=x/4;
% contourf(t,x,shrinf,20);
% colorbar;
% set(gca,'FontSize',25);
% % colormap((cool));
% % xlim([0 100]);
% ylim([-.3 .3]);
% xlabel('% Stenosis by Diameter','FontSize',35);
% ylabel('Normalized Axial Location','FontSize',35);
% title('Normalized Shear Rate','FontSize',35);
% grid on;

st=[0 50 75 85 90 93 95 98 99];
QQN=[1.00E+00 1.00E+00 9.50E-01 3.52E-01 1.53E-01 7.19E-02 3.43E-02 3.68E-03
3.46E-04];
gould=1.1-(0.016)*st+(0.00063)*(st.^2)-(0.0000057)*(st.^3);

figure
plot(st,QQN,'o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidth',
5,'LineStyle','-','Color','k')
set(gca,'FontSize',35);
pos = get(gca,'position');
  set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% xlim([min(t) max(t)]);
xlabel('% Stenosis','FontSize',40);
ylabel('Q/Q_N','FontSize',40);
grid on
hold on
E=.013*ones(length(st),1)
% errorbar(st,gould,E)
plot(st,gould,'k','LineWidth',4,'LineStyle','-.')
legend('Simulation','Gould "74')


% st=[0 50 75 85 90 93 95 98 99];
% Re=[228.9597427 457.9194854 870.0470222 537.7445443 351.3592629 235.3352903
157.0663835 42.07135272 7.922007097];
% figure
% plot(st,Re,'o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15)
% set(gca,'FontSize',25);
% pos = get(gca,'position');
%   set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% % xlim([min(t) max(t)]);
% xlabel('% Stenosis','FontSize',35);
% ylabel('Re_D Throat','FontSize',35);
% grid on
```

```
% % hold on
% % plot(st,gould,'k','LineWidth',4,'LineStyle','-.')
% % legend('Simulation','Gould "74')

fid = fopen(['E:\shr_stdy2\shear_exp\scale_4mm_final_drop\sten00-v'],'r');
shrdat = textscan(fid,'%f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
shrdat=cell2mat(shrdat);
vx=(shrdat(:,3))/max(shrdat(:,3));
fclose(fid);
figure
vm=(2.e-06)/(pi*(max(shrdat(:,3))^2));
plot(2.*(1-(vx).^2),vx,'k','LineWidth',5,'LineStyle','-')
hold on
fid = fopen(['E:\shr_stdy2\shear_exp\scale_4mm_final_drop\sten50-v'],'r');
shrdat = textscan(fid,'%f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
shrdat=cell2mat(shrdat);
vx=(shrdat(:,3))/max(shrdat(:,3));
fclose(fid);
vm=(2.e-06)/(pi*(max(shrdat(:,3))^2));
plot(shrdat(:,4)./vm,vx,'k','LineWidth',5,'LineStyle','--')
fid = fopen(['E:\shr_stdy2\shear_exp\scale_4mm_final_drop\sten95-v'],'r');
shrdat = textscan(fid,'%f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
shrdat=cell2mat(shrdat);
vx=(shrdat(:,3))/max(shrdat(:,3));
fclose(fid);
vm=(6.85874e-08)/(pi*(max(shrdat(:,3))^2));
plot(shrdat(:,4)./vm,vx,'k','LineWidth',5,'LineStyle','-.')
fid = fopen(['E:\shr_stdy2\shear_exp\scale_4mm_final_drop\sten99-v'],'r');
shrdat = textscan(fid,'%f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
shrdat=cell2mat(shrdat);
vx=(shrdat(:,3))/max(shrdat(:,3));
fclose(fid);
vm=(7.89409e-10)/(pi*(max(shrdat(:,3))^2));
% vm=max(shrdat(:,4))./2;
plot(shrdat(:,4)./vm,vx,'k','LineWidth',5,'LineStyle',':')
%
% close all
% px=csaps(shrdat(:,4)/mean(shrdat(:,4)),vx,1.1,shrdat(:,4)/mean(shrdat(:,4)));
% plot(shrdat(:,4)/mean(shrdat(:,4)),vx,'k','LineWidth',5,'LineStyle',':')
% hold on
% plot(shrdat(:,4)/mean(shrdat(:,4)),px,'r','LineWidth',5,'LineStyle',':')

set(gca,'FontSize',35);
% xlim([min(t) max(t)]);
set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
xlabel('v/v_m_e_a_n','FontSize',40);
```

```
ylabel('r/r_0','FontSize',40);
grid on
legend('0% Stenosis','50% Stenosis','95% Stenosis','99% Stenosis')
%%%%%%%%%%%%%%%%%%%%%%%%%bar_graph%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%
% figure
% b=[403195;161023;459067;403190];
% bar(b)
% boo=xticklabel_rotate([1:4],45,{'4 mm          ';'16 mm         ';'4 mm Rough    ';'4 mm
Pulsatile'},'FontSize',25)
% ylabel('Maximum Shear Rate (1/s)','FontSize',35);
% set(gca,'FontSize',25);
% set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure
st=[50 75.1 85 90 93 95 97 98 99];
b=[50000 271979 349020 383293 401021 403195 393792 424772 308111];
plot(st,b,'o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidth',5,'Li
neStyle','-','Color','k')
xlim([75 100]);
set(gca,'FontSize',35);
pos = get(gca,'position');
  set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% xlim([min(t) max(t)]);
xlabel('% Stenosis','FontSize',40);
ylabel('Shear Rate (1/s)','FontSize',40);
grid on
hold on
st=[75.1 90 95 98 99];
b=[190279 226497 231764 219973 179737];
plot(st,b,'d','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidth',5,'Li
neStyle','-.','Color','k')
st=[50 75.1 85 90 93 95 97 98 99];
b=[13727 123333.5 152191 163986 164193 161023 126006 107133 76780];
plot(st,b,'s','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidth',5,'Li
neStyle',':','Color','k')
legend('Case 1: 4 mm, dp=100 mmHg, Resistance','Case 2: 4 mm, dp=50 mmHg','Case 3:
16 mm, dp=100 mmHg, Resistance')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%
figure
st=[50 75 85 90 93 95 97 98 99];
b=[50000 271979 349020 383293 401021 403195 393792 424772 308111];
```

```
plot(st,b,'o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidth',5,'Li
neStyle','-','Color','k')
xlim([75 100]);
ylim([0 500000]);
n=get(gca,'Ytick');
set(gca,'yticklabel',sprintf('%g |',n'));
% set(gca,'yTickLabel',['200,000';'10 ';'100'])
set(gca,'FontSize',35);
pos = get(gca,'position');
  set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
xlabel('% Stenosis','FontSize',40);
ylabel('Shear Rate (1/s)','FontSize',40);
grid on
hold on


st=[0 5 10  15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 92 95 97 98 99];
b=[656252.1348 707117.2582 761169.7885 818113.483 877412.3248 938331.3451
999747.1389 1060198.107 1117755.694 1170268.446 1215003.044 1248991.826
1269036.648 1271749.631 1253684.076 1210998.396 1139327.451 1032050.486
877048.532 796150.6422 642796.2651 504612.3513 414697.4136 295112.7483];

x=sym('x');
s0=sym('s0');
mu=3.8e-9;%kg/micron/s
dp=0.0133322368;% (kg / (s^2)) / micron
r0=1500;%microns
L=4000;%microns
r=2*(r0-s0*r0*.25*((1-cos(2*pi*(x-L)/L))^2));
resist=2.24e-13;%kg/micron^4/s
g=int(1/r,x,0,L);
g=double(subs(g,s0,st/100));
% g=g+(60);
g=(2*g/mu+pi*(r0^3)*.25.*resist);

b=(dp./(g))
plot(st,b,'k','LineWidth',5,'LineStyle','--')


st=[50 75 85 90 93 95 97 98 99];
b=[13727 123334 152191 163986 164193 161023 126006 107133 76780];
plot(st,b,'s','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidth',5,'Li
neStyle',':','Color','k')


x=sym('x');
```

```
s0=sym('s0');
mu=3.8e-9;%kg/micron/s
dp=0.0133322368;% (kg / (s^2)) / micron
r0=1500;%microns
L=16000;%microns
r=2*(r0-s0*r0*.25*((1-cos(2*pi*(x-L)/L))^2));
g=int(1/r,x,0,L)+(60);

st=[0 5 10  15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 92 95 97 98 99];
b=[164061.9608 176781.4506 190293.4994 204525.2988 219357.9912 234582.8363
249932.5144 265041.8368 279445.6467 292567.1116 303746.164 312247.9564
317261.7298 317937.4079 313415.7331 302743.8904 284826.6886 258012.6215
219267.8821 199037.6606 160701.6397 126151.3751 103674.3534 73780.35683];
% b=double(dp./(2*mu.*subs(g,s0,st/100)))
plot(st,b,'k','LineWidth',5,'LineStyle','-.')

% b=[0 330982.0579 179281.948 129972.2866 106448.6566 93431.50663 85874.54654
81670.8974 79836.49248 79914.53402 81752.5683 85425.88119 91241.70569
99819.98572 112297.4839 130804.1093 159672.985 209085.8524 309668.8193
385683.5386 614680.9647 1022827.649 1533474.045 3066027.914];
% plot(st,b,'k','LineWidth',4,'LineStyle','--')
%
% b=[0.00000E+00 2.62464E+05 1.59407E+05 1.20057E+05 1.00187E+05
8.88956E+04 8.22656E+04 7.85858E+04 7.70362E+04 7.72325E+04 7.90484E+04
8.25544E+04 8.80206E+04 9.59794E+04 1.07381E+05 1.23930E+05 1.48876E+05
1.88976E+05 2.59788E+05 3.03417E+05 3.87431E+05 4.19664E+05 3.88131E+05
2.92428E+05];
% plot(st,b,'k','LineWidth',4,'LineStyle','-')
% legend('4 mm Porous-Jump','R1','R2','R1&R2')
legend('Case 1: 4 mm, dp=100 mmHg, Resistance','4 mm Viscous Resistance
(Poiseuille)','Case 3: 16 mm, dp=100 mmHg, Resistance','16 mm Viscous Resistance
(Poiseuille)')

%centerline velocity
figure
st=[75 85 90 93 95 97 98 99];
b=[4.85 5.1 5.23 5.33 5.38 5.25 4.64 2.18];
b2=[271979 349020 383293 401021 403195 393792 424772 308111];


[AX,H1,H2]=plotyy(st,b2,st,b)
set(AX(1),'FontSize',35);
set(AX(2),'FontSize',35);
set(AX(1),'XColor','k','YColor','k')
set(AX(2),'XColor','k','YColor','k')
```

```
% set(get(H2),'FontSize',35);
set(get(AX(2),'Ylabel'),'String','Maximum Centerline Velocity (m/s)','FontSize',40)
set(get(AX(1),'Ylabel'),'String','Maximum Shear Rate (1/s)','FontSize',40)
set(get(AX(1),'Xlabel'),'String','% Stenosis','FontSize',40)
set(get(AX(2),'Xlabel'),'String','% Stenosis','FontSize',40)
set(H2,'Marker','^','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidt
h',5,'LineStyle','--','Color','k')
% % xlim([min(t) max(t)]);
set(H1,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidt
h',5,'LineStyle','-','Color','k')
grid on
% title('Max Thrombus Growth w.r.t. Time','FontSize',20);
legend([H1 H2],'Shear Rate (1/s)','Centerline Velocity')
pos = get(AX,'position');
  set(AX,'position',[0.1893    0.2000    0.68    0.7250]);
% pos = get(AX(2),'position');
%   set(AX(2),'position',[0.1893    0.2000    0.7157    0.7250]);

%
plot(st,b,'o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidth',5,'Li
neStyle','-','Color','k')


% set(gca,'FontSize',35);

% xlabel('% Stenosis','FontSize',35);
% ylabel('Maximum Centerline Velocity (m/s)','FontSize',35);
% grid on


%Shear integration
figure
st=[0 50 60 70 75 85 90 93 95 97 98 99];
b=[24044.58599 4.35E+03 3.09E+03 1.83E+03 1.46E+03 1.11E+03 8.77E+02 7.69E+02
6.27E+02 4.55E+02 4.53E+02 2.41E+02];
b2=[71754.74293 4.75E+04 9.98E+04 2.35E+05 3.91E+05 3.39E+05 2.97E+05
2.62E+05 2.30E+05 1.77E+05 1.38E+05 4.89E+04]/(10^4.6);

[AX,H1,H2]=plotyy(st,b2,st,b)
set(AX(1),'FontSize',35);
set(AX(2),'FontSize',35);
set(AX(1),'XColor','k','YColor','k')
set(AX(2),'XColor','k','YColor','k')
set(get(AX(2),'Ylabel'),'String','Shear History (Eqn 2.9)','FontSize',40)
set(get(AX(1),'Ylabel'),'String','Shear History (dynes^2^.^3-s/mm^4^.^6)\n (Eqn
2.8)','FontSize',40)
```

```
set(get(AX(1),'Xlabel'),'String','% Stenosis','FontSize',40)
set(get(AX(2),'Xlabel'),'String','% Stenosis','FontSize',40)
set(H2,'Marker','^','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidt
h',5,'LineStyle','--','Color','k')
set(H1,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidt
h',5,'LineStyle','-','Color','k')
grid on
hold on
% h=plot([0 100],[3.6e6 3.6e6])
% legend([H1 H2 h],'Equation 4','Equation 5','Hellum''s limit Equation 5')
legend([H1 H2],'Equation 2.8','Equation 2.9')
pos = get(AX,'position');
  set(AX,'position',[0.1893    0.2000    0.68    0.7250]);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%555
st=[0 40 50 60 70 75 85 90 93 95 97 98 99];
st2=[75 85 90 93 95 97 98 99];


QQN=[1 0.804766511 0.727024888 0.530472254 0.348078445 0.251780846
0.09427491 0.040981444 0.019171577 0.009146073 0.002775481 0.000967373
8.7943E-05];
QQN2=[0.246416322 0.085656495 0.034473661 0.01459854 0.005980125 0.001191628
0.000263829 0];


stbaumgart=[77.6 68.4 66.8 65.4 63.9 61.3 60 60 56.4 54.2 53.1 52 51 50 49 48 47.1 44.3
38.4 33.7 29.3 29.3 29.3 22.5];
baumgart=[0.49 0.53 0.53 0.61 0.74 0.7 0.74 0.78 0.78 0.79 0.95 0.93 0.91 0.88 0.81 0.82
0.83 0.83 0.98 0.85 0.96 0.99 0.99 0.99];

stg=[50 50 50 50 50 60 60 70 70 70 70];
g=[0.845 0.845 0.845 0.786 0.786 0.704 0.66 0.445 0.445 0.35 0.35];

stgg=[94 88.5 85 82 80];
gg=[0.3 0.175 0.275 0.575 0.15];

figure
plot(st,QQN,'o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidth',
5,'LineStyle','-','Color','k')
set(gca,'FontSize',35);
pos = get(gca,'position');
  set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% xlim([min(t) max(t)]);
```

```
xlabel('% Stenosis by Diameter','FontSize',40);
ylabel('FFR','FontSize',40);
grid on
hold on
plot(stbaumgart,baumgart,'^','MarkerSize',15,'MarkerEdgeColor','k','MarkerFaceColor','k'
)
plot(stg,g,'s','MarkerSize',15,'MarkerEdgeColor','k','MarkerFaceColor','k')
plot(stgg,gg,'d','MarkerSize',15,'MarkerEdgeColor','k','MarkerFaceColor','k')
legend('Our Simulation','Baumgart "98','Shalman "01','Tron "95')
%
plot(st2,QQN2,'^','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidt
h',5,'LineStyle','-','Color','k')

st=[16 4 0.14];
QQN=[126006 393792 610639];

figure
plot(st,QQN,'o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidth',
5,'LineStyle','-','Color','k')
set(gca,'FontSize',35);
pos = get(gca,'position');
  set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% xlim([min(t) max(t)]);
xlabel('Stenosis Feature Length Scale (mm)','FontSize',40);
ylabel('Shear Rate (1/s)','FontSize',40);
grid on
```

Appendix D: Post Processing Images (Chapter 3)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Analyze images to get thrombus deposition
%David Bark
%11/11/10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all
clear java
% y='C:\Documents and Settings\Dave\Desktop\Stenosis B10 24 Sept 08';
nam=[3,240]; %min and max images to evaluate  240 180
%21
cfd=0;
% limg=248;
y='E:\2009\C26_04Sep09\C26_04Sep09_12_19_jpegs\C26_04Sep2009_12_19_'; %file location and number prefix
xlsr= xlsread('c:\SA2\C26_04Sep2009_12_19.xls','C26_04Sep2009_12_19');
k=[94 248 300 474];
% k=[1 50 150 200 250 300 350 400 450 500 550 600];
% k=[1 50 75 100 125 150 175 225 275 308];BC9_22Jul2009_09_55_000735.jpeg
% k=[450 500 600 700 800 850 900];
y_nam='c:\\SA2\\C26_04Sep2009_12_19_\\';
 vel=0.00452707394;
thresh=.185%.175;%.05
if nam(2)>152
   thresh=.109+(9.997e-7)*nam(2)^2+(2.878e-5)*nam(2)+.01887%.175;%.05.12 %.1635
%     thresh=(9.95E-04)*nam(2) - 4.32E-02
%     thresh=.13
else
   thresh=.109;
end
thresh=.109;
thresh_ref=.5;

imskip=1;

threshi=15;%17 n/a 20 25 30
n2=1.45;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Open up reference image
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y_place=y;
```

```
k=1;
ref=0;
for i=(nam(1)):(nam(1)+5)
   if ((i<10)&&(i>=1))
      img= (imread([y '00000' num2str(i) '.jpeg']));
   elseif ((i<100)&&(i>=10))
      img= (imread([y '0000' num2str(i) '.jpeg']));
   elseif ((i<1000)&&(i>=100))
      img= (imread([y '000' num2str(i) '.jpeg']));
   elseif ((i<10000)&&(i>=1000))
      img= (imread([y '00' num2str(i) '.jpeg']));
   end
   for j=1:3
      img(:,:,j)=fliplr(img(:,:,j));
   end
   %     ref.(na(k))=(double(img));
   %     k=k+1;
   %     ref=img+ref;
   ref=ref+double(img);
end
ref=uint8(ref./5);
ddd=ref;
[thet fac xmin xmax ymin ymax mm]=len_pix(ref,thresh_ref);%Determining cropping
parameters
dood=imcrop(((double((rgb2gray(ref)))./double(max(max(rgb2gray(ref))))),[0 0
size(ref,2) ymax+ymin]);
% [a b]=min(dood);
thet=thet-.0055%+.0018
% [fit1 gof]= fit([1:length(b)]',b','poly1','Robust','on');
% thet=-fit1.p1


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Finding boundaries for reference image
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
si=size(ref);
ref=imcrop(ref,[fac fac si(2)-2*fac si(1)-2*fac]); %crop image
ref = imrotate(ref,thet*180/pi); %rotate image
[di]=(double((rgb2gray(ref)))./double(max(max(rgb2gray(ref))))); % normalize image
[pita thresh]=edge(di,'log',.0003); % obtain images for reference image
boundaries = bwboundaries(pita); %obtain boundaries after edge detection
m=1000; % high number to filter boundary choices
bound=[];
for i=1:length(boundaries) %Determining largest boundary region
```

```
    b=cell2mat(boundaries(i));
    if (max(size(b))>m)
        bound=[bound i];
    end
end
for i=1:length(bound)
    figure
    b = boundaries{bound(i)};
    b = sortrows(b, 1);%Sorting to align vertical points relative to axial location - only 2
allowed
    b = sortrows(b, 2);
    plot(b(:,2),b(:,1),'.')
end
%%%%%%%%%%%%%%%%%%%
%%%choose plots that contain inner boundary of stenosis%%%%%%
b = boundaries{bound(3)};%45
b = sortrows(b, 1);%Sorting to align vertical points relative to axial location - only 2
allowed
b = sortrows(b, 2);
g = boundaries{bound(4)};
g = sortrows(g, 1);%Sorting to align vertical points relative to axial location - only 2
allowed
g = sortrows(g, 2);
figure
gg=fnval(b(:,2),csaps(g(:,2),g(:,1)));
plot(b(:,2),mean([b(:,1)';gg']),'.')
figure
plot(b(:,2),gg','.')
hold on
plot(b(:,2),b(:,1)','r.')
mid=mm+5;%356 %pick mid point.. sometiems the automated midpoint, mm, is
inaccurate
%%%%%%%%%%%%%%%%%%%
%%%choose plots that contain outer boundary of stenosis%%%%%%
gg = boundaries{bound(2)};
gg = sortrows(gg, 1);%Sorting to align vertical points relative to axial location - only 2
allowed
gg = sortrows(gg, 2);
gg1 = boundaries{bound(5)};
gg1 = sortrows(gg1, 1);%Sorting to align vertical points relative to axial location - only 2
allowed
gg1 = sortrows(gg1, 2);
figure
plot(gg(:,2),gg(:,1)','.')
hold on
plot(gg1(:,2),gg1(:,1)','r.')
```

```
x=[xmin:(xmax+xmin)];
gg=fnval(x,csaps(gg(:,2),gg(:,1))); %get all boundaries the same size
b=fnval(x,csaps(b(:,2),b(:,1)));
g=fnval(x,csaps(g(:,2),g(:,1)));
figure
imshow(di)
hold on
plot(x,gg,'r-')
plot(x,mid*ones(size(x)),'y')
plot(x,g,'b-')
plot(x,b,'g-')
ymin=round(min(g)); % cropping for min y
ymax=round(max(b)-ymin); % cropping for max y
b=b-ymin; %obtain one inner boundary
g=g-ymin; %obtain other inner boundary
x=x-xmin;
R=gg-mid; %obtain outer boundary for refraction correction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%pick a pixel resolution by dividing outer diameter by the amount the
%%%%%%outer diameter spans
ll=1500/(max(R));
ll=1500/274;
ddd2=imcrop(ref,[xmin ymin xmax ymax]);%cropped RGB image
ddd3=imcrop(di,[xmin ymin xmax ymax]);%cropped normalized gray image
r=0.5.*(b-g);%inner radius
lbound=b';%%%%%%bounds to use when picking new boundaries of images later than
reference
ubound=g';%%%%%%bounds to use when picking new boundaries of images later than
reference
[mmm ind]=min(r);%obtain middle axial point
pooy=round(mid-ymin);%mid radius indice
poox=ind(1,1);%mid axis indice
clear mmm;
x=x-x(ind(1,1));% set x=0 at stenosis peak
R=1500*ones(size(R));
r=ll.*r;% scale radius by real length
r0=r;%%% original radius
figure
plot(x.*ll,r)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% update refraction numbers
r1=real(r-R.*(1-r.^2./R.^2).^(1./2).*tan(asin(r./R)-asin(20/29.*r./R)));
hold on
plot(x.*ll,r1,'r')
r2=real(r-1500.*(1-r.^2./1500.^2).^(1./2).*tan(asin(r./1500)-asin(20/29.*r./1500)));
plot(x.*ll,r2,'g')
```

```
plot(x.*ll,R,'y')
% axis equal
figure
imshow(ddd2);
hold on
plot((x-min(x)),(mid-ymin)*ones(size(x)),'y')
plot((x-min(x)),b,'.g','LineWidth',1);
plot((x-min(x)),g,'.g','LineWidth',1);
y=ll*abs([1:size(ddd3,1)]-pooy)'*(ones(1,size(ddd3,2)));
y0=y;
rr=(ones(size(ddd3,1),1))*r1;
RR=(ones(size(ddd3,1),1))*R;
r=r1';
length_blood=1+(rr.*cos(asin(y./RR)-asin(y./n2./RR)+asin(y./rr./n2))./cos(asin(y./RR)-
asin(y./n2./RR)+asin(y./rr./n2)-asin(5./7.*y./rr)))./750;%length of vessel observed
through
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Determine growth per image
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r2=r;
r=ll.*(b-mid+ymin);
r_top=[real(r-R.*(1-r.^2./R.^2).^(1./2).*tan(asin(r./R)-asin(20/29.*r./R)))]';
r=ll.*(mid-g-ymin);
r_bot=[real(r-R.*(1-r.^2./R.^2).^(1./2).*tan(asin(r./R)-asin(20/29.*r./R)))]';
rr=[r1'];
r=r2;
% r_top
d1=double(rgb2gray(ddd2));
d1=1000+d1-mean(mean(d1([1:100],[(poox-100):(poox+100)])));
for iv=(nam(1)+1):nam(2)
y=y_place;
% if 1
%     iv=nam(2);
   if iv<=(nam(1)+2)%picking 5 images for averaging
      k=[(nam(1)):(nam(1)+4)];
   else iv==(nam(1)+3)
      k=[(iv-2):(iv+2)];
   end
   imga=0;
   for lo=k%%%averaging over 5 images
      if ((lo<10)&&(lo>=1))
         img= (imread([y '00000' num2str(lo) '.jpeg']));
```

```
        elseif ((lo<100)&&(lo>=10))
            img= (imread([y '0000' num2str(lo) '.jpeg']));
        elseif ((lo<1000)&&(lo>=100))
            img= (imread([y '000' num2str(lo) '.jpeg']));
        elseif ((lo<10000)&&(lo>=1000))
            img= (imread([y '00' num2str(lo) '.jpeg']));
        end
        for j=1:3
            img(:,:,j)=fliplr(img(:,:,j));
        end
        imga=imga+double(img);
    end
    img=imcrop(uint8(imga./5),[fac fac si(2)-2*fac si(1)-2*fac]);%averaging and cropping
    img = imrotate(img,thet*180/pi);%rotating image
    ddd=imcrop(img,[xmin ymin xmax ymax]);%refined cropping
%     figure
%     surf(double(rgb2gray(ddd))./max(max(double(rgb2gray(ddd)))),'EdgeColor','none')
%                 set(gca,'FontSize',30);
%             set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
%     xlabel('Pixel Number','FontSize',35);
%     ylabel('Pixel Number','FontSize',35);
%     zlabel('Normalized Light Intensity','FontSize',35);
%



    did=imsubtract(ddd,ddd2);%subtracting current image from orignal image

    d2=double(rgb2gray(ddd));
    d2=1000+d2-mean(mean(d2([1:100],[(poox-100):(poox+100)])));
    img=d2-d1;

%     img=double(rgb2gray(did));%converting to gray scale in double form
    middl=fnval(csaps(x,img(pooy,:),.001),x);%picking axial location
    img=img-ones(size(y0,1),1)*middl.*length_blood;
    imgg=double((img)>threshi);

     H = fspecial('gaussian',80,3);
    gra=imfilter(real(ddd(:,:,2)),H,'symmetric')
    [imggg thresh]=edge(gra,'sobel');%did(:,:,2)rgb2gray
    BW2 = bwareaopen(imggg, 10);
%     figure
%     imshow(BW2)
    imggg=BW2;
    img=imgg+imggg;
%     img=imgg;
```

```
    for i=1:length(x)
        lb=1000000;
        ub=0;
        for ii=1:pooy
            if ((img(ii,i)>.5))
                ub=ii;
            end
        end
        ii=length(img(:,1));
        while ii>pooy
            if ((img(ii,i)>.5))
                lb=ii;
            end
            ii=ii-1;
        end
        for ii=1:length(img(:,1))
            if((ii<=ubound(i,1))||(ii>=lbound(i,1))||(ii>=lb)||(ii<=ub))
                img(ii,i)=(1>0);
            end
        end
    end
    img=1-img;
    H = fspecial('motion',7,0);
    img=imfilter(real(img),H,0);
    img=img>.5;
    [b g] = boundd2 (img);
    b(b(:,2)<=(xmin+8),:)=[];
    g(g(:,2)<=(xmin+8),:)=[];
    b(b(:,2)>=(xmax-8),:)=[];
    g(g(:,2)>=(xmax-8),:)=[];
    b = interp1(b(:,2),b(:,1),x-min(x),'nearest','extrap')
    g = interp1(g(:,2),g(:,1),x-min(x),'nearest','extrap')
    y=ll.*.5.*(b-g)';
    rr=[rr r.*sin(asin(y./R')-asin(y./n2./R')+asin(y./r./n2))-r.*cos(asin(y./R')-
asin(y./n2./R')+asin(y./r./n2)).*tan(asin(y./R')-asin(y./n2./R')+asin(y./r./n2)-
asin(5./7.*y./r))];
    y=ll.*(mid-g-ymin)';
    r_bot=[r_bot r.*sin(asin(y./R')-asin(y./n2./R')+asin(y./r./n2))-r.*cos(asin(y./R')-
asin(y./n2./R')+asin(y./r./n2)).*tan(asin(y./R')-asin(y./n2./R')+asin(y./r./n2)-
asin(5./7.*y./r))];
    y=ll.*(b-mid+ymin)';
    r_top=[r_top r.*sin(asin(y./R')-asin(y./n2./R')+asin(y./r./n2))-r.*cos(asin(y./R')-
asin(y./n2./R')+asin(y./r./n2)).*tan(asin(y./R')-asin(y./n2./R')+asin(y./r./n2)-
asin(5./7.*y./r))];
end
```

```
figure
imshow(ddd);
hold on
plot((x-min(x)),b,'.g','LineWidth',1);
plot((x-min(x)),g,'.g','LineWidth',1);
[po]=get(gca,'Position');
annotation('arrow',[po(1)+2*po(3)/3 po(1)+po(3)/5+2*po(3)/3],[po(4)/2+po(2)
po(4)/2+po(2)],'LineWidth',10,'Color',[0.8314 0.8157
0.7843],'HeadStyle','plain','HeadWidth',50,'HeadLength',30);


figure
d1=double(rgb2gray(ddd));
d1=1000+d1-mean(mean(d1([1:100],[(poox-100):(poox+100)])));
% d1=1000+d1-max(d1(:));
plot(d1(:,poox)/1000,'linewidth',5,'LineStyle','-','color',[.0 .0 .0]);
hold on
d2=double(rgb2gray(ddd2));
% d2=1000+d2-max(d2(:));
d2=1000+d2-mean(mean(d2([1:100],[(poox-100):(poox+100)])));
plot(d2(:,poox)/1000,'linewidth',5,'LineStyle','--','color',[.0 .0 .0]);

plot([lbound(poox) lbound(poox)],[0 2],'linewidth',5,'LineStyle','-.','color',[.0 .0 .0]);
plot([g(poox) g(poox)],[0 2],'linewidth',5,'LineStyle',':','color',[.0 .0 .0]);

plot([ubound(poox) ubound(poox)],[0 2],'linewidth',5,'LineStyle','-.','color',[.0 .0 .0]);
plot([b(poox) b(poox)],[0 2],'linewidth',5,'LineStyle',':','color',[.0 .0 .0]);


set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
grid on
xlabel('Pixel #','FontSize',35);
ylabel('Light Intensity (I/I_0)','FontSize',35);
legend('Time: 0s','Time: 415s','Initial Edge','Thrombus Edge');



figure
d1=double(rgb2gray(ddd));
d1=1000+d1-mean(d1(1,:));
% d1=1000+d1-max(d1(:));
plot(d1(1,:)/1000)
hold on
d2=double(rgb2gray(ddd2));
% d2=1000+d2-max(d2(:));
```

```
d2=1000+d2-mean(d2(1,:));
plot(d2(1,:)/1000,'r')
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
grid on
xlabel('Pixel #','FontSize',35);
ylabel('Light Intensity (I/I_m_a_x)','FontSize',35);


% d1=imsubtract(d1-max(d1(:)),d2-max(d2(:)));
% % d1=rgb2gray(d1);
% figure
% plot(d1(:,poox))



r=r/ll;


figure
plot(r*ll-rr(:,2))


doodl=rr(:,50:length(rr(1,:)));
dat=ll*r*ones(size(rr(1,:)))-rr

dat=imfilter(real(dat),H,0);

H = fspecial('gaussian',80,3);
dat=imfilter(real(rr),H,'symmetric')

fresult= csaps(xlsr(:,2),xlsr(:,4),.05);
fresult=fnder(fresult);
fresult=fnval(xlsr(:,2),fresult);

fresult=((fresult([nam(1):nam(2)])/ 0.001073)/(pi*.75^2))/1000;
timq=xlsr([nam(1):nam(2)],2);
xlsr_tim=timq;

figure
plot(timq,fresult,'b');
hold on
timq= csaps(timq,fresult,.0001,timq);
plot(xlsr_tim,timq,'r');
pres=xlsr([nam(1):nam(2)],3)*133.322;
pres=csaps(xlsr_tim,pres,.0001,xlsr_tim);
```

```
timqq=timq*(pi*.00075^2)

timm=xlsr_tim-min(xlsr_tim);
tim=timm;

% dat=double(dat>0).*dat;
% xy={x,tim};
j=1;
clear iv;
```

Appendix E: Evaluate Dat of Processed Images

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Evaluate post-processed data pertaining to images
%David Bark
%11/11/10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all
clear all


n=['c04';'c23';'c24';'c26';'c30';'b77';'b74';'b82';'bc9'];
% n=['c24'];
% n=['c04';'c23';'c24';'c26';'c29';'c30';'b77';'b74'; 'b82'; 'bc9'; 'b68'; 'd02'];
% n=['c04';'c23';'c24';'c26';'c29';'c30'];
% n=['c04';'c23';'c24';'c26';'c30';'b74';'bc9';'b82'];
% n=['c04';'c23';'c24';'c26';'c30'];
% n=['c04';'c23';'c24';'c30';'b74'];
% n=['c04';'c26'];
expr.c04.file='C:\SA2\fix12_final_C4_22Jul200912_42.mat';
expr.c23.file='C:\SA2\fix7_final_C23_22Jul2009_10_24.mat';
expr.c24.file='C:\SA2\fix4_final_C24_26Aug2009_11_04.mat';
expr.c26.file='C:\SA2\fix5_final_C26_04Sep2009_12_19.mat';
expr.c29.file='C:\SA2\new_t_C29_28Sep2009_11_06.mat';
expr.c30.file='C:\SA2\fix8_final_C30_28Sep2009_10_44.mat';
expr.b77.file='C:\SA2\fix2_final_B77_29Jun2009_10_08.mat';
expr.b74.file='C:\SA2\fix2_final_B74_06Jul2009_09_48.mat';
expr.b82.file='C:\SA2\fix3_final_B82_15Jun2009_11_12.mat';
expr.bc9.file='C:\SA2\fix_alt_final_BC9_22Jul2009_09_55.mat';
expr.b68.file='C:\SA2\b68_17Mar2010_10_48.mat';
expr.d02.file='C:\SA2\d2_17Mar2010_15_12.mat';


expr.c04.y_nam='C:\SA2\C4_22Jul2009_12_42_\';
expr.c23.y_nam='C:\SA2\C23_22Jul2009_10_24_\';
expr.c24.y_nam='C:\SA2\C24_26Aug2009_11_04_\';
expr.c26.y_nam='C:\SA2\C26_04Sep2009_12_19_\';
expr.c29.y_nam='C:\SA2\C29_28Sep2009_11_06_\';
expr.c30.y_nam='C:\SA2\C30_28Sep2009_10_44_\';
expr.b77.y_nam='C:\SA2\B77_29Jun2009_10_08_\';
expr.b74.y_nam='C:\SA2\B74_06Jul2009_09_48_\';
expr.b82.y_nam='C:\SA2\B82_15Jun2009_11_12_\';
expr.bc9.y_nam='C:\SA2\BC9_22Jul2009_09_55_\';
expr.b68.y_nam='C:\SA2\b68_17Mar2010_10_48_\';
expr.d02.y_nam='C:\SA2\d2_17Mar2010_15_12_\';
```

```
expr.c04.k=[97 100:25:225 248 275:25:425 434 442 450 458 466 474];
expr.c04.k=[97 100:25:225 248 275:25:425 434 442 450 458];
expr.c04.k=[97 100:25:225 248 280 320 340:10:400 420:10:430 450:10:460];
expr.c04.k=[97 340 380 400 420 440 460];
expr.c04.k=[94 100 110 130:10:460];
expr.c04.k=[100:10:280 300:10:460];
expr.c23.k=[65 75:25:250 268 275 282 284 292 300 308 317 325 334 342 350];
expr.c23.k=[65 75:25:250 260:10:350];
expr.c23.k=[66 100:10:150 180 200:10:250 270:10:350];
expr.c23.k=[65 70:10:310 330 340 350];
expr.c23.k=[70:10:350];
expr.c24.k=[35 50:25:200 210 212 218 220 225 230 232 238 244 250];
expr.c24.k=[34 40:10:230 242];
expr.c24.k=[40:10:240];
expr.c26.k=[4 50 75 100 125 150 163 175 180:10:250];
expr.c26.k=[4 50 75 100:10:190 200:10:240];
expr.c26.k=[3 10:10:240]
expr.c26.k=[10:10:240];
expr.c29.k=[28 50 75 100 125 150 175 200 212 225 238 242];
expr.c29.k=[28 120 180 200];
expr.c30.k=[56 100 110:5:135 150 200 225 250 275 284 292 300 308 316 325 333 342
350 361];
expr.c30.k=[56 60:10:350];
expr.c30.k=[55 60:10:230 250:10:350 359];
expr.c30.k=[60:10:350];
% expr.b77.k=[94 125:25:650];
expr.b77.k=[94 100:10:650];
expr.b74.k=[100:25:250 270:5:360 375:25:650];
expr.b74.k=[60:10:650];
expr.b82.k=[26 30:10:430];
expr.bc9.k=[60:10:650];%56
% expr.b68.k=[50:50:850];
expr.b68.k=[50:50:700];
expr.d02.k=[30 60 150:50:450 460];
cas=[5 6 7 8 9 10 4 1 2 3];
cas=[1 2 3 4 5 6 7 8 9 10];
% cas=[1 2 3 4 5 6 7 8 9 10 11 12];

normaliz= [.25 .42 .265 .514 .803 .54 0 0 0 0]*1e12;%flow rates for syringe pump
expreriments microns^3/minute.265->.276
% normaliz= [.25 .803]*1e12;%flow rates for syringe pump expreriments
microns^3/minute.265->.276
% normaliz= [0 0 0 0 0 0];%flow rates for syringe pump expreriments
for i=[1:length(n(:,1))]
   eval(['expr.', n(i,:),'.normaliz=',num2str(normaliz(i)),';'])
```

```
    eval(['expr.', n(i,:),'.cas=',num2str(cas(i)),';'])
end

thresho=.001; %degree of growth interpolation
phs1tim=10;
phs2tim=10;
% n=['bc9'];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
gim=[];
imm=[];
rimm=[];
timm=[];
xim=[];
are_hh=[];
are_ll=[];
v_h=[];
v_l=[];
sv_h=[];
sv_l=[];
dpv_ll=[];
dpv_hh=[];
spv_ll=[];
spv_hh=[];
r_0=[];
s_0=[];
r_280=[];
s_280=[];
r_200=[];
s_200=[];
r_120=[];
s_120=[];
for iv=[1:length(n(:,1))]
    % iv=2;
    eval(['load(expr.',n(iv,:),'.file);'])
    x=x';
    xy={x,tim};

    eval(['expr.', n(iv,:),'.x=[',num2str(x'),'];'])
    eval(['expr.', n(iv,:),'.tim=[',num2str(tim'),'];'])
    eval(['k=[expr.', n(iv,:),'.k];'])
    eval(['y_nam=[expr.', n(iv,:),'.y_nam];'])
    eval(['normaliz=[expr.', n(iv,:),'.normaliz];'])
    daty=dat;
    if normaliz==0
        Q=timq*(pi*.00075^2)*6e19*.002/8;
```

```
      QQ=(Q*ones(size(x')))';
   else
      QQ=normaliz.*ones(size(dat))*.002/8;
   end
   if n(iv,:)=='b77'
      tim=tim+110;
   end
   %     H = fspecial('motion',20,45);
   Q=timq*(pi*.00075^2);
   H = fspecial('motion',20,0);
   %%%%%good%%%%%%%%%%%%
   %     im = imfilter(dat,H,'replicate');%filtering high freq from data noise
   %     im=dat;
   %     rim = imfilter(rr,H,'replicate');%filtering high freq from data noise
   %     rim=rr;
   %
   %     H = fspecial('gaussian',80,3);
   %     dat=ll*r*ones(size(rr(1,:)))-imfilter(real(rr),H,'symmetric');
   %     rim=imfilter(real(rr),H,'symmetric');

   im=dat;
   rim=dat;
   dat=dat(:,1)*ones(size(dat(1,:)))-dat+1e-8;





%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
   %     figure
   %     xy={x,tim};
   %     contourf(tim,x.*ll./750,im./750,20)
   %     ylim([-2 2]);
   %     set(gca,'FontSize',30);
   %     set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
   %     xlabel('Time (s)','FontSize',35);
   %     ylabel('Axial Location/R_0','FontSize',35);
   %     title('Thrombus Growth/R_0','FontSize',35);
   %     grid on;
   %     colorbar;

   %%%%%graph%%%%%%%%%%%%%%
   %     plot(im(700,:))
```

```
%    hold on
%    plot(dat(700,:),'r')
%
%    plot(im(650,:))
%    plot(im(750,:))
%    plot(im(600,:))
%
%    plot(dat(650,:),'r')
%    plot(dat(750,:),'r')
%    plot(dat(600,:),'r')
%    figure
%    contourf(im)
%    figure
%    contourf(dat)

%%%%%%%%%%Evaluate Shear in comp domain%%%%%%%%%
shrinf=[];
shrhist=[];
press=[];
presss=[];
for j=k
   fid = fopen([y_nam,num2str(j),'-'],'r');
   shrdat = textscan(fid,'%f %f %f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
   shrdat=cell2mat(shrdat);
   shrx=1000*(shrdat(:,2));
   shry=1000*(shrdat(:,3));
   pres=(shrdat(:,4));
   presss=[presss max(pres)];
   pres=diff(pres)./diff(sqrt(shrx.^2+shry.^2));
   pres=[pres(1);pres];
   shrm=(shrdat(:,5));
   shrm=-shrm.*sign((shrdat(:,6)));
   fclose(fid);
   %        fit1=malowess(ll.*x, shrinf, 'Robust', 'on')';
   %        shrinf=fit1';

   %        sr=csaps(shrx,shrm,.9995,ll.*x/1000);

   sr=fnval(ll.*x/1000,csaps(shrx,shrm));%x in mm
   shrinf=[shrinf sr];%x in mm
   press=[press csaps(shrx,pres,.9995,ll.*x/1000)];
end
figure
plot(abs(diff(mean(shrinf))./mean(shrinf(:,[1:(size(shrinf,2)-1)])),'.')
%    shrinf=imfilter(shrinf,H,'symmetric');
if normaliz>0
```

```
        hyde=xlsr([nam(1):nam(2)],3)*133.322./(.5*1025*(normaliz*(1.6667e-
20)/(pi*.00075^2)).^2);
        hyd=presss./(.5*1025*(normaliz*(1.6667e-20)/(pi*.00075^2)).^2);

        hyde=xlsr([nam(1):nam(2)],3)*133.322;
        hyd=presss;
    else
        fresult= csaps(xlsr(:,2),xlsr(:,4),.05);
        fresult=fnder(fresult);
        fresult=fnval(xlsr(:,2),fresult);
        fresult=((fresult([nam(1):nam(2)])/ 0.001073)/(pi*.75^2))/1000;
        hyde=xlsr([nam(1):nam(2)],3)*133.322./(.5*1025*(fresult).^2);
        hyd=presss./(.5*1025*(fresult(k-nam(1)+1)').^2);

        hyde=xlsr([nam(1):nam(2)],3)*133.322;
        hyd=presss;
    end

    % %%%%%%%%%%%%%%%%%%%%%%%%%%%%
    figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
    %     axes1 = axes('FontSize',30,'Parent',figure11);
    plot(x*ll./750,rr(:,1)./750,'LineWidth',3,'LineStyle','-','Color',[.1 .1 .1]);
    hold on
    plot(x*ll./750,rr(:,length(rr(1,:)))./750,'LineWidth',3,'LineStyle','--','Color',[.3 .3 .3]);
    axis equal
    xlim([-3 3]);
    legend('Initial Radius','Case Termination Radius');
    grid on
    set(gca,'FontSize',30);
    set(gca,'position',[0.1893    0.2000    0.69    0.7]);
    annotation(figure11,'textbox','Position',[0.37 -0.05 0.25
0.16],'LineStyle','none','FitHeightToText','off','String',{'Normalized Axial Location
(z/r_0)'},'Fontsize',35);
    xlabel('Normalized Axial Location (z/r_0)','Fontsize',35);
    ylabel('Normalized Radius (r/r_0)','Fontsize',35);
    ylim([0 1]);


    %
    %     [po]=get(gca,'Position');
    %     annotation('arrow',[po(1)+po(3)/3 po(1)+po(3)/3+po(3)/3],[po(4)/2+po(2)
po(4)/2+po(2)],'HeadWidth',50,'LineWidth',5);
    title(['Case:',num2str(eval(['expr.', n(iv,:),'.cas']))],'Fontsize',35)
```

```
    figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
    y_sr=fnval(x,csaps(x,shrinf(:,1),.0001));
    %     figure
    semilogy(x*ll./750,abs(y_sr),'LineWidth',3,'LineStyle','-.','Color',[.2 .2 .2]);
    hold on
    y_sr=fnval(x,csaps(x,shrinf(:,length(k)),.01));
    semilogy(x*ll./750,abs(y_sr),'LineWidth',3,'LineStyle',':','Color',[.4 .4 .4]);

semilogy(x(y_sr<0)*ll./750,abs(y_sr(y_sr<0)),'ko','MarkerSize',10,'MarkerFaceColor','k')
;
    grid on
    set(gca,'FontSize',30);
    set(gca,'position',[0.1893    0.2000    0.69    0.7]);
    title(['Case:',num2str(eval(['expr.', n(iv,:),'.cas']))],'Fontsize',35)
    xlabel('Normalized Axial Location (z/r_0)','Fontsize',35);
    ylabel('Shear Rate (1/s)','FontSize',35);
    ylim([10 10^6]);
    xlim([-3 3]);

    legend('Initial Shear Rate','Case Termination Shear Rate','Negative Shear');%,'boxoff');

    [po]=get(gca,'Position');
    annotation('arrow',[po(1)+po(3)/3 po(1)+po(3)/3+po(3)/3],[po(4)/2+po(2)
po(4)/2+po(2)],'HeadWidth',50,'LineWidth',5);

    if iv<=5
        x2=ll.*x';
        rim2=r_bot;
        rim3=r_top;


        if min(x2)>-2000
            x2=[-3000 x2];
            rim2=[rim2(1,:);rim2];
            rim3=[rim3(1,:);rim3];
        end
        if max(x2)<2000
            x2=[x2 3000];
            rim2=[rim2;rim2(length(rim2(:,1)),:)];
            rim3=[rim3;rim3(length(rim3(:,1)),:)];
        end
        x2=x2./750;
        rim2=rim2./750;
        rim3=rim3./750;
        figure1=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
        %     plot(x.*ll,rim(:,length(tim)),'LineWidth',5,'LineStyle','-','Color','b')
```

```
ax1=fill([x2 fliplr(x2)],[-rim3(:,1)' fliplr(rim2(:,1)')],[.95 .95 .95])%'r'
hold on

ax2=fill([min(x2) x2 max(x2)],[2000 rim2(:,length(tim))' 2000],[.90 .9 .9])
ax3=fill([min(x2) x2 max(x2)],[-2000 -rim3(:,length(tim))' -2000],[.9 .9 .9])
%    alpha(ax1,.00001);
%    alpha(ax2,.00001);
%    alpha(ax3,.00001);

ax4=fill([min(x2) x2 max(x2)],[2000 rim2(:,1)' 2000],'w')
ax5=fill([min(x2) x2 max(x2)],[-2000 -rim3(:,1)' -2000],'w')
%    plot(x,rim(:,1),'LineWidth',5,'LineStyle','--','Color','k')

set(gca,'FontSize',30);
set(gca,'position',[0.2    0.2000    0.73    0.73]);
set(gca,'TickDir','out');
%    xlim([-3 3])
yl=ylim;
axis equal
xlim([-3 3])
ylim([-1 3])



%    axis equal
%    xlim([-3 3])
[po]=get(gca,'Position')
%    [po1]=get(gca,'InnerPosition')
%    annotation('arrow',[po(1)+5*po(3)/6 po(1)+3*po(3)/3],[po(4)/2+po(2)
po(4)/2+po(2)],'HeadWidth',50,'LineWidth',5);
title(['Case:',num2str(eval(['expr.', n(iv,:),'.cas']))])
hold on
h1 = gca;

set(h1,'YMinorTick','off','XTickMode','manual','XTick',[],'YTickMode','manual','YTick',[
]);
h2 = axes('Position',get(h1,'Position'));
y_sr=fnval(x,csaps(x,shrinf(:,1),.0001));
ax6=semilogy(x*ll./750,abs(y_sr),'LineWidth',5,'LineStyle','-','Color','k');
hold on
y_sr=fnval(x,csaps(x,shrinf(:,length(k)),.01));
ax7=semilogy(x*ll./750,abs(y_sr),'LineWidth',5,'LineStyle','--','Color','k');

ax8=semilogy(x(y_sr<0)*ll./750,abs(y_sr(y_sr<0)),'ko','MarkerSize',10,'MarkerFaceColo
r','k');
grid on
```

```
        ylim([10 10^6]);

set(h2,'Color','none','YMinorTick','off','YMinorGrid','off','YTickLabel',{'10';'100';'1,000';'
10,000';'100,000';'1,000,000'});%,'position',get(h1,'position')
        set(h2,'XLim',get(h1,'XLim'),'Layer','top','FontSize',30)
        pol=plotboxpos(h1);
        set(h2,'position',pol)
        xlabel('Normalized Axial Location (z/r_0)','Fontsize',35);
        ylabel('Shear Rate (1/s)','FontSize',35);
        legend([ax6,ax7,ax8],'Initial Shear Rate','Case Termination Shear Rate','Negative
Shear');%,'boxoff');
        %     set(gcf,'PaperPositionMode','auto')

        %     set(h1,'YLim',[-1 3])
        %     set(h1,'XLim',[-3 3])
    end

    %%%%%%%%%%%%%%%%%%%%%
    %       figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
    %     %     axes1 = axes('FontSize',30,'Parent',figure11);
    %     [ax1]=plot(x*ll./750,rr(:,1)./750,'LineWidth',3,'LineStyle','-','Color',[.1 .1 .1]);
    %     hold on
    %     [ax2]=plot(x*ll./750,rr(:,length(rr(1,:)))./750,'LineWidth',3,'LineStyle','--
','Color',[.3 .3 .3]);
    %     xlim([-3 3]);
    %     %     legend('Initial Radius','Case Termination Radius','boxoff');
    %     grid on
    %     set(gca,'FontSize',30);
    %     set(gca,'position',[0.1893    0.2000    0.69    0.7]);
    %     annotation(figure11,'textbox','Position',[0.37 -0.05 0.25
0.16],'LineStyle','none','FitHeightToText','off','String',{'Normalized Axial Location
(z/r_0)'},'Fontsize',35);
    %     ylabel('Normalized Radius (r/r_0)');
    %     ylim([0 1]);
    %     h1 = gca;
    %     h2 = axes('Position',get(h1,'Position'));
    %
    %     y_sr=fnval(x,csaps(x,shrinf(:,1),.0001));
    %     %     figure
    %     [ax3]=semilogy(x*ll./750,abs(y_sr),'LineWidth',3,'LineStyle','-.','Color',[.2 .2 .2]);
    %     hold on
    %     y_sr=fnval(x,csaps(x,shrinf(:,length(k)),.01));
    %     [ax4]=semilogy(x*ll./750,abs(y_sr),'LineWidth',3,'LineStyle',':','Color',[.4 .4 .4]);
    %
[ax5]=semilogy(x(y_sr<0)*ll./750,abs(y_sr(y_sr<0)),'ko','MarkerSize',10,'MarkerFaceCol
or','k');
```

```
%     title(['Case:',num2str(eval(['expr.', n(iv,:),'.cas']))],'Fontsize',35)
%
%     %     ylim([-20000 60000]);
%     %
set(h2,'YAxisLocation','right','Color','none','XTickLabel',[],'YTick',[],'YTickLabel',[]);
%     set(h2,'YAxisLocation','right','Color','none','YMinorTick','off');
%     ylabel('Shear Rate (1/s)','FontSize',35);
%     ylim([10 10^6]);
%     set(h2,'XLim',get(h1,'XLim'),'Layer','top','FontSize',30)
%     xlim([-3 3]);
%     h3=gca;
%     legend([ax1,ax2,ax3,ax4,ax5],'Initial Radius','Case Termination Radius','Initial
Shear Rate','Case Termination Shear Rate','Negative Shear');%,'boxoff');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%




% for j=k
%         fid = fopen([y_nam,num2str(j)],'r');
%         shrdat = textscan(fid,'%f %f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
%         shrdat=cell2mat(shrdat);
%         shrx=1000*(shrdat(:,2));
%         shry=1000*(shrdat(:,3));
%         shrm=(shrdat(:,4));
%         shrm=-shrm.*sign((shrdat(:,5)));
%         fclose(fid);
%         %         fit1=malowess(ll.*x, shrinf, 'Robust', 'on')';
%         %         shrinf=fit1';
%         sr=csaps(shrx,shrm,.9995,ll.*x/1000);
%
%         %         shrinf=[shrinf fnval(ll.*x/1000,csaps(shrx,shrm))];%x in mm
%         shrinf=[shrinf sr];%x in mm
%         if j==k(1)
%             shrst=zeros(size(sr));
%             shrhist=[shrhist shrst];
%         else
%             shrst=shrst+(tim(j-nam(1)+1)-tim(j-1-nam(1)+1)).*sr.^2.3;
%             shrhist=[shrhist shrst];
%         end
%     end
% press=ones(size(shrinf));
```

```
%     s=(4.16666665).*4./pi./(shry.^3);
%     figure
%     plot(shrx,s)
%     hold on
%     plot(shrx,shrm,'r')
%     figure
%     plot(shrx,s./shrm)
%%%%%%%%%%shear gradient%%%%%%
H = fspecial('gaussian',80,3);
%     H = fspecial('motion',60,0);
shrinff=imfilter(real(shrinf),H,'replicate')
sgrad=[];
for i=1:length(k)
   sgrad=[sgrad fnval(x,csaps(x,shrinff(:,i),.0001))];
end
%     figure
%     plot(sgrad(:,1))
shrinff=sgrad;


sgrad=[];
for i=1:length(k)
   sgrad=[sgrad (diff(shrinff(:,i))./diff(ll.*x))];
end
sgrad=[sgrad;sgrad(length(sgrad(:,1)),:)];
%     sgrad=shrinff./sgrad;
%     shrinf=shrinff;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     figure
%     contourf(tim(k-nam(1)+1),ll.*x/750,shrinf,20);
%     set(gca,'FontSize',30);
%     set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
%     ylim([-2 2]);
%     xlabel('Time (s)','FontSize',35);
%     ylabel('Axial Location/R_0','FontSize',35);
%     title('Shear Rate (1/s)','FontSize',35);
%     grid on;
%     colorbar
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     %Plot shear contour
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    %     numberOfColors = 6; % Choose the number of colors
    %     nc = numberOfColors-1; % This value will be used in the code below to set the
number of colors as desired
    %     figure
    %     % shrinf=shrinf./max(shrinf(:));
    %     %     contourf(tim(k-nam(1)+1),ll.*x/750,shrinf,[0 500 2000 5000 50000]);
    %     contourf(tim(k-nam(1)+1),ll.*x/750,shrinf,[0 500 2000 5000 50000]);
    %     set(gca,'FontSize',35);
    %     set(gca,'position',[0.3    0.2000    0.6    0.7250]);
    %     % colorbar;
    %     colormap(flipud(gray));
    %     % xlim([0 100]);
    %     ylim([-2 2]);
    %     xlabel('Time (s)','FontSize',40);
    %     ylabel('(Axial Position)/r_0','FontSize',40);
    %     title('Shear Rate (1/s)','FontSize',40);
    %     grid on;
    %     caxis([-1000 10000])
    %     %specifiy colormap scaling
    %     H=cbarf(shrinf,[0 500 2000 5000 50000],'vertical','nonlinear');
    %     set(H,'YTickLabel',{'0','500','2,000','5,000','50,000'},'FontSize',35)
    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %     %Plot pressure contour
    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %     numberOfColors = 6; % Choose the number of colors
    %     nc = numberOfColors-1; % This value will be used in the code below to set the
number of colors as desired
    %     figure
    %     % shrinf=shrinf./max(shrinf(:));
    %     %     contourf(tim(k-nam(1)+1),ll.*x/750,shrinf,[0 500 2000 5000 50000]);
    %     contourf(tim(k-nam(1)+1),ll.*x/750,press);
    %     set(gca,'FontSize',35);
    %     set(gca,'position',[0.3    0.2000    0.6    0.7250]);
    %     % colorbar;
    %     colormap(flipud(gray));
    %     % xlim([0 100]);
    %     ylim([-2 2]);
    %     xlabel('Time (s)','FontSize',40);
    %     ylabel('(Axial Position)/r_0','FontSize',40);
    %     title('Pressure Differential (Pa/m)','FontSize',40);
    %     grid on;
    %     %specifiy colormap scaling
```

```
%     colorbar
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     %Plot growth contour
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     numberOfColors = 6; % Choose the number of colors
%     nc = numberOfColors-1; % This value will be used in the code below to set the number of colors as desired
%     figure
%     % shrinf=shrinf./max(shrinf(:));
%     %     contourf(tim(k-nam(1)+1),ll.*x/750,shrinf,[0 500 2000 5000 50000]);
%     contourf(tim,ll.*x/750,im,[5 15 25 35 45 55 65]);
%     set(gca,'FontSize',35);
%     set(gca,'position',[0.3    0.2000    0.6    0.7250]);
%     % colorbar;
%     colormap(flipud(gray));
%     % xlim([0 100]);
%     ylim([-2 2]);
%     xlabel('Time (s)','FontSize',40);
%     ylabel('(Axial Position)/r_0','FontSize',40);
%     title('Growth (microns)','FontSize',40);
%     grid on;
%     caxis([0 65])
%     %specifiy colormap scaling
% %      H=cbarf(im,[5 15 25 35 45 55 65],'vertical','nonlinear');
%     set(H,'YTickLabel',{'5','15','25','35','45','55','65'},'FontSize',35)
%%%%%%%%%%%%%%%%%%%%%%%%%dep rate%%%%%%
dppp2=[];
for i=1:length(x)
   doo=(-60.*(.8/8.8).*(diff(rim(i,:))./diff(tim')));
   for j=1:(length(tim)-1)
      if (abs(doo(j))>100000)%6
         if j==1
            doo(j)=doo(j+1);
         else
            doo(j)=doo(j-1);
         end
      end
   end
   dppp2=[dppp2;doo];
end
dppp2=[dppp2 dppp2(:,length(dppp2(1,:)))];
H = fspecial('motion',60,0);
```

```
dppp3 = imfilter(dppp2,H,'replicate');
%     dppp3 = dppp2;


% %%other method
% dppp=[];
% depr=[];
% fresult =  csaps(xy,-60.*(.8/8).*rim,thresho,xy);
% for i=1:length(x)
%     fresult2=csaps(tim,fresult(i,:),thresho);
%     depr=[depr;fnval(tim,fresult2)'];
%     fresult2=fnder(fresult2);
%     fresult2=fnval(tim,fresult2);
%     dppp=[dppp;fresult2'];
% end
% figure
% contourf (dppp)
%

%%%%Calculate time initation phs1%%%%%%

xyk={x,tim(k-nam(1)+1)};
shrinfint=csaps(xyk,shrinf,.5,xy);
init=[];
rinit=[];
shrinit=[];
xinit=[];
% ti=[];
tj=[];
QQ=[];
for j=1:length(x)
   for i=2:length(tim)
      if((dat(j,i)>phs1tim)&&(dat(j,i-1)<phs1tim))
         %           ti=[ti i];
         tj=[tj j];
         init=[init tim(i)];
         xinit=[xinit x(j)];
         rinit=[rinit rr(j,i)];
         %           QQ=[QQ Q(i)*60000000];
         break
      end
   end
end
%%%%Calculate time initation phs2%%%%%%
init2=[];
xinit2=[];
```

```
    for j=1:length(x)
       for i=2:length(tim)
          if((dat(j,i)>phs2tim)&&(dat(j,i-1)<phs2tim))
             init2=[init2 tim(i)];
             xinit2=[xinit2 x(j)];
             break
          end
       end
    end


    %%%%%%%%%%%%%%%%%%%%convert to shear data%%%%%%%%%%
    separationint =  csaps(xinit2,init2,.05,x);

separationint=(x>max(xinit2)).*100000000+(x<min(xinit2)).*100000000+separationint;
    grwthterm=[];
    shrterm=[];
    rimm=[];
    xterm=[];
    sinit=[];
    spinit=[];
    sxinit=[];
    srinit=[];
    sgradd=[];
    spinit=[];
    tjj=[];
    ti=[];
    grwim=[];
    grwdat=[];
    vel2=[];
    dppp4=zeros(size(dat(:,1)));
    for jj=2:length(k)
       dppp4=[dppp4,-6.*(im(:,k(jj)-nam(1)+1)-im(:,k(jj-1)-nam(1)+1))./(tim(k(jj)-
nam(1)+1)-tim(k(jj-1)-nam(1)+1))];
    end


    for jj=1:length(k)
       shrterm=[shrterm shrinf((tim(k(jj)-nam(1)+1)>=separationint),jj)'];
       grwthterm=[grwthterm dppp3((tim(k(jj)-nam(1)+1)>=separationint),k(jj)-
nam(1)+1)'];
       grwdat=[grwdat dat((tim(k(jj)-nam(1)+1)>=separationint),k(jj)-nam(1)+1)'];
       %        grwthterm=[grwthterm dppp4((tim(k(jj)-nam(1)+1)>=separationint),jj)'];
       spinit=[spinit press((tim(k(jj)-nam(1)+1)>=separationint),jj)'];%ll.*x(j)/1000];
       sgradd=[sgradd sgrad((tim(k(jj)-nam(1)+1)>=separationint),jj)'];
       grwim=[grwim im((tim(k(jj)-nam(1)+1)>=separationint),k(jj)-nam(1)+1)'];
```

```
        rimm=[rimm rim((tim(k(jj)-nam(1)+1)>=separationint),k(jj)-nam(1)+1)'];
        xterm=[xterm ll.*x((tim(k(jj)-nam(1)+1)>=separationint))'];%ll.*x(j)/1000];
        sinit=[sinit init((tim(k(jj)-nam(1)+1)==init))];
        sxinit=[sxinit ll.*xinit((tim(k(jj)-nam(1)+1)==init))];
        tjj=[tjj tj((tim(k(jj)-nam(1)+1)==init))];
        srinit=[srinit rinit((tim(k(jj)-nam(1)+1)==init))];%ll.*x(j)/1000];
        ti=[ti tim(k(jj)-nam(1)+1).*ones(1,sum(tim(k(jj)-nam(1)+1)>=separationint))];
        if iv>=6
           vel2=[vel2 vel(k(jj)-nam(1)+1).*ones(1,sum(tim(k(jj)-
nam(1)+1)>=separationint))];
        end
    end
    %     shrterm=shrterm(grwim>=phs2tim);
    %     grwthterm=grwthterm(grwim>=phs2tim);
    %     spinit=spinit(grwim>=phs2tim);
    %     sgradd=sgradd(grwim>=phs2tim);
    %     grwim=grwim(grwim>=phs2tim);
    %     rimm=rimm(grwim>=phs2tim);
    %     xterm=xterm(grwim>=phs2tim);
    %     ti=ti(grwim>=phs2tim);

    %     shrterm=shrterm(grwthterm~=0);
    %     grwthterm=grwthterm(grwthterm~=0);
    %     sgradd=sgradd(grwthterm~=0);
    %     xterm=xterm(grwthterm~=0);
    %     sum(grwthterm==0)
    sshrinit=[];
    for jj=1:length(sinit)
        sshrinit=[sshrinit shrinf(tjj(jj),tim(k-nam(1)+1)==sinit(jj))];%ll.*x(j)/1000];
    end

    %     eval(['expr.', n(iv,:),'.im=[',num2str(im(:)'),';'])
    %     eval(['expr.', n(iv,:),'.gim=[',num2str(dppp3(:)'),';'])
    alph= fittype('60*(1e-8)*(2.75e8)*k*c/(1+1.8575*k/((a*x+1.6e-
9)*(x/(.00025*(a*x+1.6e-9)))^(1/3)))');
    %     poos=shrterm(shrterm>0);
    %     poog=grwthterm(shrterm>0);
    %     poog=poog(poos<=100000);
    %     poos=poos(poos<=100000);
    %     [fit1 gof]= fit(poos',poog',alph,'StartPoint',[1e-8 1 .012],'Lower',[0 0
0],'Robust','on');
    %     fdata = feval(fit1,shrterm);
    %     fit1=malowess(shrterm, grwthterm, 'Robust', 'on');
    %     fdata = fit1';

    %
```

```
spinit=(spinit.*(2.5e-3)+shrterm.*.0038);
%    %%%%%change%%%%%%%%%
%    %    sgradd=sgradd(shrterm>=0);
%    %    rimm=rimm(shrterm>=0);
%    %    xterm=xterm(shrterm>=0);
%    %    grwthterm=grwthterm(shrterm>=0);
%    %    spinit=spinit(shrterm>=0);
%    %    shrterm=shrterm(shrterm>=0);
%
%        [po po1]=sort(shrterm);
%        grwthterm=grwthterm(po1);
%        shrterm=shrterm(po1);
%        spinit=spinit(po1);
%
%        % [fit1 gof]=
fit(shrterm(shrterm<=1800)',grwthterm(shrterm<=1800)','poly1','Robust','on');
%        % [fit2 gof]=
fit(shrterm(shrterm>5000)',grwthterm(shrterm>5000)','poly1','Robust','on');
%        % fdata1=feval(fit1,shrterm');
%        % fdata2=feval(fit2,shrterm');
%        % fdata=zeros(size(grwthterm))';
%        % fdata(fdata1<=fdata2)=fdata1(fdata1<=fdata2);
%        % fdata(fdata2<fdata1)=fdata2(fdata2<fdata1);
%
% %        [fit1 gof]= fit(shrterm',grwthterm',alph,'StartPoint',[1e-8 1 .012],'Lower',[0 0
0],'Robust','on');
% %        fdata = feval(fit1,shrterm);
%        fit1=malowess(shrterm', grwthterm', 'Robust', 'on');
%        fdata = fit1;
%
%
%
%
%        %  any(diff(sign(feval(fit1,shrterm')-feval(fit2,shrterm'))))
%
%        I = (fdata' - grwthterm) > 1.*std(grwthterm);
%        outliers = excludedata(shrterm,grwthterm,'indices',I);
%        I = -(fdata' - grwthterm) > 1.*std(grwthterm);
%        outliers2 = excludedata(shrterm,grwthterm,'indices',I);
%
%        figure
%        plot(shrterm,fdata','r-
',shrterm,grwthterm,'k.',shrterm(outliers),grwthterm(outliers),'m*',shrterm(outliers2),grwt
hterm(outliers2),'g*')
%
%        figure
```

```
    %       contourf(tim,x.*ll,im,30)
    %       hold on
    %
plot(ti(outliers),xterm(outliers),'Linestyle','none','Marker','*','MarkerEdgeColor','m','Mark
erFaceColor','m','MarkerSize',10);
    %
plot(ti(outliers2),xterm(outliers2),'Linestyle','none','Marker','x','MarkerEdgeColor','g','Mar
kerFaceColor','g','MarkerSize',10);
    %       ylim([-750 750]);
    %       set(gca,'FontSize',30);
    %       set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
    %       xlabel('Time (s)','FontSize',35);
    %       ylabel('Axial Location','FontSize',35);
    %       title('Thrombus Growth','FontSize',35);
    %       grid on;
    %       colorbar;
    %       outliers=logical(1-((outliers+outliers2)>0));
    %       %    outliers=logical(ones(1,length(outliers)));
    %       %    contourf(
    %       %    plot(fit1,'r-',poos,poog,'k.',outliers,'m*')
    %       %    hold on
    %       %    plot(fit2,'c--')

    outliers=logical(ones(size(shrterm)));

    eval(['expr.', n(iv,:),'.init=[',num2str(sinit),'];'])
    eval(['expr.', n(iv,:),'.grwdat=[',num2str(grwdat),'];'])
    eval(['expr.', n(iv,:),'.hyde=[',num2str(hyde'),'];'])
    eval(['expr.', n(iv,:),'.hyd=[',num2str(hyd),'];'])
    eval(['expr.', n(iv,:),'.tim_shr=[',num2str(tim(k-nam(1)+1)'),'];'])
    eval(['expr.', n(iv,:),'.p=[',num2str(spinit),'];'])
    eval(['expr.', n(iv,:),'.xinit=[',num2str(sxinit),'];'])
    eval(['expr.', n(iv,:),'.rinit=[',num2str(srinit),'];'])
    eval(['expr.', n(iv,:),'.srinit=[',num2str(sshrinit),'];'])
    eval(['expr.', n(iv,:),'.shear=[',num2str(shrterm(outliers)),'];'])
    eval(['expr.', n(iv,:),'.grw=[',num2str(grwthterm(outliers)),'];'])
    eval(['expr.', n(iv,:),'.sgrad=[',num2str(sgradd(outliers)),'];'])
    eval(['expr.', n(iv,:),'.xgrw=[',num2str(xterm(outliers)),'];'])
    eval(['expr.', n(iv,:),'.ti=[',num2str(ti(outliers)),'];'])
    eval(['expr.', n(iv,:),'.rim=[',num2str(rimm(outliers)),'];'])
    if n(iv,1)=='c'
       vel1=vel.*ones(size(grwdat));
    else
       vel1=vel2;
    end
    eval(['expr.', n(iv,:),'.vel=[',num2str(vel1),'];'])
```

```
    imm=[imm im(:)'];
    gim=[gim dppp3(:)'];

    are_h=zeros(7,5);
    are_l=zeros(7,5);
    dpv=zeros(7,5);
    spv=zeros(7,5);
    tbin_h=[190 210 290 310 390 410 490 510 590 610];
    tbin_l=[40 60 90 110 140 160 190 210 240 260];
    tim_h=[200 300 400 500 600];
    tim_l=[50 100 150 200 250];
    xbin=[-1.75 -1.25 -.75 -.25 .25 .75 1.25 1.75];
    xb=[-1.5 -1 -.5 0 .5 1 1.5];

    art=[];
    arm=[];
    volt=pi.*ll.*(repmat(r.*ll,1,length(tim)).^2-rr.^2);
    vo=volt((x>-300)&(x<300),:);
    vo=vo-(repmat(vo(:,2),1,length(tim)));
    figure
    plot(tim,sum(vo));
    hold on
%     [til xi]=sort(tim);
%     vo=sum(vo(xi));
    volt=volt-(repmat(volt(:,2),1,length(tim)));
    art=2.*pi.*r.*ll.*ll;

    tim_v=[0:10:600];
    [nn,xbins]=histc(x*ll./750,xbin);%xbins is the bin location
    [nn,vbins]=histc(x*ll./750,[-2 2]);%xbins is the bin location
    [nn,tvbins]=histc(tim,tim_v);%xbins is the bin location
    stv=mean(shrinf(vbins==1,:));
    stv=csaps(tim(k-nam(1)+1),stv,.9995,tim_v);
    stv(tim_v>max(tim(k-nam(1)+1)))=0;

    plot(tim,sum(volt(vbins==1,:)),'r');

    if
(strcmp(n(iv,:),'b74')||strcmp(n(iv,:),'bc9'))%((strcmp(n(iv,:),'b77')||strcmp(n(iv,:),'b74'))||s
trcmp(n(iv,:),'bc9'))
        for j=1:length(xb)
            arm=sum(volt(j==xbins,:))./sum(art(j==xbins));
            tbin=tim_h;
            [nn,tbins]=histc(tim,tbin_h);%xbins is the bin location
            for i=1:2:length(tbin_h)
```

```
            are_h(j,(i+1)./2)=mean(arm(i==tbins));
            dpv(j,(i+1)./2)=mean2(dppp3(xbins==j,i==tbins));
            spv(j,(i+1)./2)=mean2(shrinfint(xbins==j,i==tbins));
         end
      end
      are_hh=[are_hh are_h];
      dpv_hh=[dpv_hh dpv];
      spv_hh=[spv_hh spv];
      tv=sum(volt(vbins==1,:));
      tvv=[];
      for i=1:length(tim_v)
         tvv=[tvv mean(tv(i==tvbins))];
      end
      v_h=[v_h;tvv];
      sv_h=[sv_h;stv];
   elseif
(strcmp(n(iv,:),'c04')||strcmp(n(iv,:),'c23')||strcmp(n(iv,:),'c24')||strcmp(n(iv,:),'c26')||strcm
p(n(iv,:),'c29')||strcmp(n(iv,:),'c30'))
      for j=1:length(xb)
         arm=sum(volt(j==xbins,:))./sum(art(j==xbins));
         tbin=tim_l;
         [nn,tbins]=histc(tim,tbin_l);%xbins is the bin location
         for i=1:2:length(tbin_l)
            are_l(j,(i+1)./2)=mean(arm(i==tbins));
            dpv(j,(i+1)./2)=mean2(dppp3(xbins==j,i==tbins));
            spv(j,(i+1)./2)=mean2(shrinfint(xbins==j,i==tbins));
         end
      end
      are_ll=[are_ll are_l];
      dpv_ll=[dpv_ll dpv];
      spv_ll=[spv_ll spv];
      tv=sum(volt(vbins==1,:));
      tvv=[];
      for i=1:length(tim_v)
         if (tv(i==tvbins))>0
            tvv=[tvv mean(tv(i==tvbins))];
         else
            tvv=[tvv 0];
         end
      end
      v_l=[v_l;tvv];
      sv_l=[sv_l;stv];
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      shrinfint=shrinfint;
      dat_r=[];
      shrinf_r=[]
```

```
        for il=1:length(dat(1,:))
            dat_r=[dat_r;interp1(x.*ll./750,dat(:,il),[-2:.002:2])];
            shrinf_r=[shrinf_r;interp1(x.*ll./750,shrinfint(:,il),[-2:.002:2])];
        end
        dat_r=dat_r';
        shrinf_r=shrinf_r';
        shrbin=[0 10];
        [nn,xbins]=histc(tim-min(tim),shrbin);%xbins is the bin location
        r_0=[r_0 dat_r(:,xbins==1)];
        s_0=[s_0 shrinf_r(:,xbins==1)];

        shrbin=[110 130];
        [nn,xbins]=histc(tim-min(tim),shrbin);%xbins is the bin location
        r_120=[r_120 dat_r(:,xbins==1)];
        s_120=[s_0 shrinf_r(:,xbins==1)];

        shrbin=[190 210];
        [nn,xbins]=histc(tim-min(tim),shrbin);%xbins is the bin location
        r_200=[r_200 dat_r(:,xbins==1)];
        s_200=[s_0 shrinf_r(:,xbins==1)];


        shrbin=[270 290];
        [nn,xbins]=histc(tim-min(tim),shrbin);%xbins is the bin location
        r_280=[r_280 dat_r(:,xbins==1)];
        s_280=[s_0 shrinf_r(:,xbins==1)];
    end
end
%%%%%%%%%%%%%%%%%
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
%     axes1 = axes('FontSize',30,'Parent',figure11);
[ax1]=plot([-2:.002:2],mean(r_0'),'LineWidth',3,'LineStyle','-','Color',[.1 .1 .1]);
hold on
[ax2]=plot([-2:.002:2],mean(r_120'),'LineWidth',3,'LineStyle','--','Color',[.3 .3 .3]);
[ax3]=plot([-2:.002:2],mean(r_200'),'LineWidth',3,'LineStyle','-.','Color',[.5 .5 .5]);
[ax4]=plot([-2:.002:2],mean(r_280'),'LineWidth',3,'LineStyle',':','Color',[.7 .7 .7]);
xlim([-3 3]);
%     legend('Initial Radius','Case Termination Radius','boxoff');
grid on
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.69    0.7]);
annotation(figure11,'textbox','Position',[0.37 -0.05 0.25
0.16],'LineStyle','none','FitHeightToText','off','String',{'Normalized Axial Location
(z/r_0)'},'Fontsize',35);
ylabel('Thrombus Thickness (\mum)');
ylim([0 500]);
```

```
legend([ax1,ax2,ax3,ax4],'Time: 0s','Time: 120s','Time: 200s','Time: 280s');%,'boxoff');
%%%%%%%%%%%%%%%%%%
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
%     axes1 = axes('FontSize',30,'Parent',figure11);
[ax1]=semilogy([-2:.002:2],mean(s_0'),'LineWidth',3,'LineStyle','-','Color',[.1 .1 .1]);
hold on
[ax2]=semilogy([-2:.002:2],mean(s_120'),'LineWidth',3,'LineStyle','--','Color',[.3 .3 .3]);
[ax3]=semilogy([-2:.002:2],mean(s_200'),'LineWidth',3,'LineStyle','-.','Color',[.5 .5 .5]);
[ax4]=semilogy([-2:.002:2],mean(s_280'),'LineWidth',3,'LineStyle',':','Color',[.7 .7 .7]);
xlim([-3 3]);
%     legend('Initial Radius','Case Termination Radius','boxoff');
grid on
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.69    0.7]);
annotation(figure11,'textbox','Position',[0.37 -0.05 0.25
0.16],'LineStyle','none','FitHeightToText','off','String',{'Normalized Axial Location
(z/r_0)'},'Fontsize',35);
ylabel('Shear Rate (1/s)');
%     ylim([0 500]);
legend([ax1,ax2,ax3,ax4],'Time: 0s','Time: 120s','Time: 200s','Time: 280s');%,'boxoff');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure2=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
doob=mean(v_l*10^-6);
doobs=std(v_l*10^-6);
ax1=plot(tim_v(tim_v<=260),doob(tim_v<=260),'LineWidth',5,'LineStyle','-','Color',[.6
.6 .6]);
hold on
plot(tim_v(tim_v<=260),v_l(:,(tim_v<=260))*10^-6,'LineWidth',5,'LineStyle','--
','Color',[.6 .6 .6]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure2=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
doob=mean(v_l*10^-6);
doobs=std(v_l*10^-6);
ax1=plot(tim_v(tim_v<=260),doob(tim_v<=260),'LineWidth',5,'LineStyle','-','Color',[.6
.6 .6]);
hold on
ax2=plot(tim_v(tim_v<=260),doob(tim_v<=260)+doobs(tim_v<=260),tim_v(tim_v<=26
0),doob(tim_v<=260)-doobs(tim_v<=260),'LineWidth',5,'LineStyle','-.','Color',[.6 .6 .6]);
doob=mean(v_h*10^-6);
doobs=std(v_h*10^-6)./sqrt(2);
ax3=plot(tim_v,doob,'LineWidth',5,'LineStyle','--','Color',[.4 .4 .4]);
ax4=plot(tim_v,doob+doobs,tim_v,doob-doobs,'LineWidth',5,'LineStyle','-.','Color',[.4 .4
.4]);
hold off
```

```
h1 = gca;
ylim([0 180]);
set(h1,'YAxisLocation','left','Color','none','YTick',[0:30:180],'YTickLabel',{'0','30','60','90
','120','150','180'});
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.6    0.7250]);
grid on
xlabel('Time (s)','FontSize',35);
ylabel('Thrombus Growth (million microns^3)','FontSize',35);

h2 = axes('Position',get(h1,'Position'));
doob=mean(sv_l);
doobs=std(sv_l)./sqrt(6);
ax5=plot(tim_v(tim_v<=260),doob(tim_v<=260),'LineWidth',5,'LineStyle','-','Color',[.2
.2 .2]);
hold on
ax6=plot(tim_v(tim_v<=260),doob(tim_v<=260)+doobs(tim_v<=260),tim_v(tim_v<=26
0),doob(tim_v<=260)-doobs(tim_v<=260),'LineWidth',5,'LineStyle','-.','Color',[.2 .2 .2]);
doob=mean(sv_h);
doobs=std(sv_h)./sqrt(2);
ax7=plot(tim_v,doob,'LineWidth',5,'LineStyle','--','Color',[0 0 0]);
ax8=plot(tim_v,doob+doobs,tim_v,doob-doobs,'LineWidth',5,'LineStyle','-.','Color',[0 0
0]);
ylim([0 24000]);
set(h2,'YAxisLocation','right','Color','none','YTick',[0:4000:24000],'YTickLabel',{'0','4,0
00','8,000','12,000','16,000','20,000','24,000'});%{'0','2,500','5,000','7,500','10,000','12,500
','15,000'}
ylabel('Shear Rate (1/s)','FontSize',35);
set(h2,'XLim',get(h1,'XLim'),'Layer','top','FontSize',30)
% legend([ax1;ax2;ax4],'Time','Shear','Initial Shear');

%     set(gca,'Visible','off');
%     alpha(ax3,0.06)
%     set(gca,'FontSize',30);
%     %     set(h3,'YAxisLocation','right','Color','none','XTickLabel',[])
%     set(h3,'XLim',get(h1,'XLim'),'Layer','bottom')
%     annotation1 = annotation(figure2,'rectangle',[0.1273 0.9296 0.08623
0.05278],'FaceColor',[1 1 1],'EdgeColor','none','LineStyle','none');
%     [po]=get(gca,'Position');
%     annotation('textarrow',[po(1)+po(3)/3 po(1)+po(3)/3+po(3)/3],[po(4)/2+po(2)
po(4)/2+po(2)],'HeadWidth',50,'LineWidth',3,'String','Flow Direction','FontSize',30);



p=.01;
gbin=zeros(7,5);
```

```
gbinn=zeros(7,5);
hig=[];
h=[];
for i=1:length(tim_h)
   for j=1:length(xb)
      rimy=[];
      for jj=[1:length(are_hh(1,:))/5]
         rimy=[rimy are_hh(j,i*jj)];
      end
      gbin(j,i)=mean(rimy);%averaged growth
      gbinn(j,i)=std(rimy);%averaged growth
      %       [h,significance,ci] = ttest2(rimy(xbins==i),rimy(xbins==(i-2)),p)
   end
end

groupnames = {'200'; '300'; '400'; '500'; '600'};
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
barweb(gbin',gbinn'./sqrt(2), [], groupnames, [], 'Time (s)', 'Thrombus Volume/Surface
Area (microns)', bone, 'y',{'-1.5'; '-1'; '-.5'; '0'; '.5'; '1'; '1.5'}, 1, 'axis')
%
% ylim([0 120]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.7157    0.7]);
annotation(figure11,'textbox','Position',[0.45 -0.05 0.25
0.16],'LineStyle','none','FitHeightToText','off','String',{'Axial Location/r0'},'Fontsize',35);



p=.01;
gbin=zeros(7,5);
gbinn=zeros(7,5);
hig=[];
h=[];
for i=1:length(tim_l)
   for j=1:length(xb)
      rimy=[];
      for jj=[1:length(are_ll(1,:))/5]
         rimy=[rimy are_ll(j,i*jj)];
      end
      gbin(j,i)=mean(rimy);%averaged growth
      gbinn(j,i)=std(rimy);%averaged growth
      %       [h,significance,ci] = ttest2(rimy(xbins==i),rimy(xbins==(i-2)),p)
   end
end
```

```
groupnames = {'50'; '100'; '150'; '200'; '250'};
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
barweb(real(gbin'),gbinn'./sqrt(6), [], groupnames, [], 'Time (s)', 'Thrombus
Volume/Surface Area (microns)', bone, 'y',{'-1.5'; '-1'; '-.5'; '0'; '.5'; '1'; '1.5'}, 1, 'axis')
%
% ylim([0 120]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.7157    0.7]);
annotation(figure11,'textbox','Position',[0.45 -0.05 0.25
0.16],'LineStyle','none','FitHeightToText','off','String',{'Axial Location/r0'},'Fontsize',35);

p=.01;
gbin=zeros(7,5);
gbinn=zeros(7,5);
hig=[];
h=[];
for i=1:length(tim_h)
   for j=1:length(xb)
      rimy=[];
      for jj=[1:length(are_hh(1,:))/5]
         rimy=[rimy .1.*are_hh(j,i*jj)];
      end
      gbin(j,i)=mean(rimy);%averaged growth
      gbinn(j,i)=std(rimy);%averaged growth
      %        [h,significance,ci] = ttest2(rimy(xbins==i),rimy(xbins==(i-2)),p)
   end
end

groupnames = {'200'; '300'; '400'; '500'; '600'};
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
barweb(gbin',gbinn'./sqrt(2), [], groupnames, [], 'Time (s)', 'Platelets/micron^2', bone,
'y',{'-1.5'; '-1'; '-.5'; '0'; '.5'; '1'; '1.5'}, 1, 'axis')
%
% ylim([0 120]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.7157    0.7]);
annotation(figure11,'textbox','Position',[0.45 -0.05 0.25
0.16],'LineStyle','none','FitHeightToText','off','String',{'Axial Location/r0'},'Fontsize',35);




p=.01;
gbin=zeros(7,5);
gbinn=zeros(7,5);
hig=[];
```

```
h=[];
for i=1:length(tim_l)
   for j=1:length(xb)
      rimy=[];
      for jj=[1:length(are_ll(1,:))/5]
         rimy=[rimy .1.*are_ll(j,i*jj)];
      end
      gbin(j,i)=mean(rimy);%averaged growth
      gbinn(j,i)=std(rimy);%averaged growth
      %        [h,significance,ci] = ttest2(rimy(xbins==i),rimy(xbins==(i-2)),p)
   end
end

groupnames = {'50'; '100'; '150'; '200'; '250'};
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
barweb(gbin',gbinn'./sqrt(6), [], groupnames, [], 'Time (s)', 'Platelets/micron^2', bone,
'y',{'-1.5'; '-1'; '-.5'; '0'; '.5'; '1'; '1.5'}, 1, 'axis')
%
% ylim([0 120]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.7157    0.7]);
annotation(figure11,'textbox','Position',[0.45 -0.05 0.25
0.16],'LineStyle','none','FitHeightToText','off','String',{'Axial Location/r0'},'Fontsize',35);




shrbin=[0 1 4.5 5.5];
[nn,xbins]=histc(imm,shrbin);%xbins is the bin location
[h,significance,ci] = ttest2(gim(xbins==1),gim(xbins==3))

shrbin=[0:.5:50];
sb=mean([shrbin([1:(length(shrbin)-1)]);shrbin([2:(length(shrbin)-0)])]);%x bin values



[nn,xbins]=histc(imm,shrbin);%xbins is the bin location
gbin=[];
gbinn=[];
sb2=[];
for i=[1:length(sb)]
   if (sum(gim(xbins==i))~=0)
      gbin=[gbin (mean(gim(xbins==i)))];%averaged growth
      gbinn=[gbinn std((gim(xbins==i)))];%averaged growth
      sb2=[sb2 sb(i)];
```

```
    end
end
figure
hh=errorbar(sb2,gbin,gbinn,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','Mar
kerSize',10,'LineWidth',3);
hold on
plot([5 5],[-6 6],'k--','LineWidth',5)
plot([15 15],[-6 6],'k--','LineWidth',5)
xlim([0 20]);
ylim([-1 5]);
errorbar_tick(hh)
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
grid on
xlabel('Thrombus Growth (microns)','FontSize',35);
ylabel('Thrombus Growth Rate (Platelets/microns^2-min)','FontSize',35);


p=.01;
gbin=zeros(7,5);
gbinn=zeros(7,5);
hig=[];
h=[];
for i=1:length(tim_l)
    for j=1:length(xb)
        rimy=[];
        for jj=[1:length(dpv_ll(1,:))/5]
            rimy=[rimy dpv_ll(j,i*jj)];
        end
        gbin(j,i)=mean(rimy);%averaged growth
        gbinn(j,i)=std(rimy);%averaged growth
        %         [h,significance,ci] = ttest2(rimy(xbins==i),rimy(xbins==(i-2)),p)
    end
end

groupnames = {'50'; '100'; '150'; '200'; '250'};
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
barweb(gbin',gbinn'./sqrt(6), [], groupnames, [], 'Time (s)', 'Growth Rate
(Platelets/\mum^2-min)', bone, 'y',{'-1.5'; '-1'; '-.5'; '0'; '.5'; '1'; '1.5'}, 1, 'axis')
handles=barweb(gbin',gbinn'./sqrt(6), [], groupnames, [], 'Time (s)', 'Growth Rate
(Platelets/\mum^2-min)', bone, 'y',{'-1.5'; '-1'; '-.5'; '0'; '.5'; '1'; '1.5'}, 1, 'axis');
%
ylim([0 3]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.69    0.7]);
```

```
annotation(figure11,'textbox','Position',[0.45 -0.05 0.25
0.16],'LineStyle','none','FitHeightToText','off','String',{'Normalized Axial Location
(z/r_0)'},'Fontsize',35);
h1 = gca;
h2 = axes('Position',get(h1,'Position'));

gbin=zeros(7,5);
gbinn=zeros(7,5);
for i=1:length(tim_l)
   for j=1:length(xb)
      rimy=[];
      for jj=[1:length(spv_ll(1,:))/5]
         rimy=[rimy spv_ll(j,i*jj)];
      end
      gbin(j,i)=mean(rimy);%averaged growth
      %        [h,significance,ci] = ttest2(rimy(xbins==i),rimy(xbins==(i-2)),p)
   end
end
gbin=gbin';
numbars = size(gbin, 2); % number of bars in a group
bar(zeros(size(gbin)), 1,'edgecolor','k', 'linewidth', 2);
hold on
for i = 1:numbars
   x =get(get(handles.bars(i),'children'), 'xdata');
   x = mean(x([1 3],:));

plot(x,gbin(:,i),'Linestyle','none','Marker','o','MarkerEdgeColor','k','MarkerFaceColor','w','
MarkerSize',20);
end
ylim([0 15000]);
set(h2,'YAxisLocation','right','Color','none','XTickLabel',[],'YTick',[-
5000:5000:20000],'YTickLabel',{'-5,000','0','5,000','10,000','15,000','20,000'});

set(h2,'YAxisLocation','right','Color','none');
ylabel('Shear Rate (1/s)','FontSize',35);
set(h2,'XLim',get(h1,'XLim'),'Layer','top','FontSize',30)



p=.01;
gbin=zeros(7,5);
gbinn=zeros(7,5);
hig=[];
h=[];
for i=1:length(tim_h)
   for j=1:length(xb)
```

```
        rimy=[];
        for jj=[1:length(dpv_hh(1,:))/5]
            rimy=[rimy dpv_hh(j,i*jj)];
        end
        gbin(j,i)=mean(rimy);%averaged growth
        gbinn(j,i)=std(rimy);%averaged growth
        %         [h,significance,ci] = ttest2(rimy(xbins==i),rimy(xbins==(i-2)),p)
    end
end

groupnames = {'50'; '100'; '150'; '200'; '250'};
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
barweb(gbin',gbinn'./sqrt(2), [], groupnames, [], 'Time (s)', 'Growth Rate
(Platelets/\mum^2-min)', bone, 'y',{'-1.5'; '-1'; '-.5'; '0'; '.5'; '1'; '1.5'}, 1, 'axis')
handles=barweb(gbin',gbinn'./sqrt(2), [], groupnames, [], 'Time (s)', 'Growth Rate
(Platelets/\mum^2-min)', bone, 'y',{'-1.5'; '-1'; '-.5'; '0'; '.5'; '1'; '1.5'}, 1, 'axis');
%
% ylim([0 120]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.69    0.7]);
annotation(figure11,'textbox','Position',[0.45 -0.05 0.25
0.16],'LineStyle','none','FitHeightToText','off','String',{'Axial
Location/r_0'},'Fontsize',35);



h1 = gca;
h2 = axes('Position',get(h1,'Position'));

gbin=zeros(7,5);
gbinn=zeros(7,5);
for i=1:length(tim_h)
    for j=1:length(xb)
        rimy=[];
        for jj=[1:length(spv_hh(1,:))/5]
            rimy=[rimy spv_hh(j,i*jj)];
        end
        gbin(j,i)=mean(rimy);%averaged growth
        %         [h,significance,ci] = ttest2(rimy(xbins==i),rimy(xbins==(i-2)),p)
    end
end
gbin=gbin';
numbars = size(gbin, 2); % number of bars in a group

bar(zeros(size(gbin)), 1,'edgecolor','k', 'linewidth', 2);
hold on
```

```
for i = 1:numbars
    x =get(get(handles.bars(i),'children'), 'xdata');
    x = mean(x([1 3],:));

plot(x,gbin(:,i),'Linestyle','none','Marker','o','MarkerEdgeColor','k','MarkerFaceColor','w','
MarkerSize',20);
end
ylim([-20000 60000]);
set(h2,'YAxisLocation','right','Color','none','XTickLabel',[],'YTick',[-
20000:10000:60000],'YTickLabel',{'-20,000','-
10,000','0','10,000','20,000','30,000','40,000','50,000','60,000'});
set(h2,'YAxisLocation','right','Color','none');
ylabel('Shear Rate (1/s)','FontSize',35);
set(h2,'XLim',get(h1,'XLim'),'Layer','top','FontSize',30)
```

Appendix F: Run CFD for Processed Images (Chapter 3)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Run CFD over processed images
%David Bark
%11/11/10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% a_f=1.2;
% a_fi=.003;
% b_f=.0075;
%
% b_f=.75;
% 30-6
% 60-6
%
% % B_f=.01;
% % m=sym('m');
% % m=solve(a_f*(a_f/(a_f-1))^m-a_f+1-B_f*a_f^m/b_f,m);
%
% nu=(b_f*a_f-a_fi)/(a_f-1)
%
% m=log(b_f/a_fi)/log(a_f)
% tott=a_fi*[1 a_f a_f^2 a_f^3 a_f^4 a_f^5]
% tott=sum(tott)
% py=sym('py');
% double(a_fi*symsum(a_f^py,0,5))
%
% a_fi=b_f*a_f-tott*(a_f-1)
% m=log(b_f/a_fi)/log(a_f)
%
% m=log(b_f/a_fi)/log((a_fi-tott)/(b_f-tott))
% a_f=(b_f/a_fi)^(1/m)

for doodlee=100:10:460
% if 1
%     doodlee=290;%430;%43065
j=doodlee-nam(1)+1;%c23 130, c4 120 get 220 c23 250  210 b74 540
vell=vel;
% vell=vel(j)

%%1%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% vel=timq(j-nam(1)+1)
fid = fopen('c:\SA2\sten.jou', 'wt');
% fprintf(fid, 'vertex create coordinates 5.365 0 0\n');
```

```
% j=2;
xyy=[min(x):((max(x)-min(x))/500):max(x)];
fresult =  csaps(x,dat(:,j)/1000,.1,xyy);
[mmm ind]=min(csaps(x,dat(:,1)/1000,.1,xyy));

le=length(fresult);
% figure
% plot(xyy,fresult)
% hold on
% xyy=x;
% fresult=rr(:,j)'/1000;
%
%
%
%
% plot(xyy,fresult,'r')
% fresult=dat(:,j)'/1000;
% plot(xyy,fresult,'g')
%
% le=length(fresult);
xyy1=ll.*xyy./1000;
fresult1=fresult;


xyy=[ll.*xyy(1)/1000-.1 ll.*xyy/1000 ll.*xyy(le)/1000+.1];
fresult=[.75 fresult .75];

%
% j=nam(2)-nam(1)+1;

le=length(fresult);


for(i=1:le)
   fprintf(fid, 'vertex create "r%i" coordinates %6.3f%6.3f%6.3f\n', i, xyy(i), fresult(i),
0.0);
end

fprintf(fid, 'edge create "sten" nurbs ');
for(i=1:(ind-round(le/8)))
   fprintf(fid, '"r%i" ', i);
end
fprintf(fid, 'interpolate\n');
fprintf(fid, 'edge create "sten1" nurbs ');
for(i=(ind-round(le/8)):(ind+round(le/8)))
   fprintf(fid, '"r%i" ', i);
```

```
end
fprintf(fid, 'interpolate\n');
fprintf(fid, 'edge create "sten2" nurbs ');
for(i=(ind+round(le/8)):le)
   fprintf(fid, '"r%i" ', i);
end
fprintf(fid, 'interpolate\n');

% fprintf(fid, 'edge create "sten" nurbs ');
% for(i=1:le)
%    fprintf(fid, '"r%i" ', i);
% end
% fprintf(fid, 'interpolate\n');

fprintf(fid, 'vertex create "x_prox_wall" coordinates %6.3f%6.3f%6.3f\n', -15*.75, .75,
0.0);
fprintf(fid, 'vertex create "x_prox_ax" coordinates %6.3f%6.3f%6.3f\n', -15*.75, 0.0,
0.0);
fprintf(fid, 'vertex create "x_dist_wall" coordinates %6.3f%6.3f%6.3f\n', (3/2)*15*.75,
.75, 0.0);
fprintf(fid, 'vertex create "x_dist_ax" coordinates %6.3f%6.3f%6.3f\n', (3/2)*15*.75, 0.0,
0.0);


fprintf(fid, 'vertex create "x_prox_ax_mid" coordinates %6.3f%6.3f%6.3f\n', xyy(1), 0.0,
0.0);
fprintf(fid, 'vertex create "x_dist_ax_mid" coordinates %6.3f%6.3f%6.3f\n', xyy(le), 0.0,
0.0);
fprintf(fid, 'vertex create "ax_ind1" coordinates %6.3f%6.3f%6.3f\n', xyy(ind-
round(le/8)), 0.0, 0.0);
fprintf(fid, 'vertex create "ax_ind2" coordinates %6.3f%6.3f%6.3f\n',
xyy(ind+round(le/8)), 0.0, 0.0);
fprintf(fid, 'edge create "wall_prox" "in" "ax_prox" "ax_mid" "ax_mid1" "ax_mid2"
"ax_dist" "out" "wall_dist" straight "r%i" "x_prox_wall" "x_prox_ax" "x_prox_ax_mid"
"ax_ind1" "ax_ind2" "x_dist_ax_mid" "x_dist_ax" "x_dist_wall"  "r%i"\n', 1,le);
fprintf(fid, 'edge create "prox_bord" straight "r%i" "x_prox_ax_mid"\n', 1);
fprintf(fid, 'edge create "dist_bord" straight "r%i" "x_dist_ax_mid"\n', le);
fprintf(fid, 'face create "prox" wireframe "wall_prox" "in" "ax_prox" "prox_bord"
real\n');
fprintf(fid, 'face create "dist" wireframe "ax_dist" "out" "wall_dist" "dist_bord"  real\n');
% fprintf(fid, 'face create "mid" wireframe "sten" "sten1" "sten2" "prox_bord"
"dist_bord" "ax_mid"  real\n');
fprintf(fid, 'face create "mid" wireframe "sten" "sten1" "sten2" "prox_bord" "dist_bord"
"ax_mid" "ax_mid1" "ax_mid2"  real\n');
% fprintf(fid, 'edge mesh "sten" "sten1" "sten2" successive ratio1 1 size 0.012\n');
% fprintf(fid, 'edge mesh "sten" successive ratio1 1 size 0.012\n');
```

```
% fprintf(fid, 'edge mesh "sten" successive ratio1 1 intervals %f\n',2*round((1215-
2300*min(fresult)/.75)/2));
% fprintf(fid, 'edge mesh "sten" successive ratio1 1 intervals 640\n');
% fprintf(fid, 'edge mesh "ax_mid" successive ratio1 1 intervals %f\n',2*round((1178-
2230*min(fresult)/.75)/2));
% len_ax=max(xyy1)-min(xyy1);
len_ax=xyy(ind-round(le/8))-xyy(2);
grid_si=.75/20;
% stret=1.005;
tott=len_ax;
b_f=grid_si;
a_fi=min(fresult)/20;
% a_fi=grid_si*stret-.5*len_ax*(stret-1)
% m=round(log(grid_si/a_fi)/log(stret))
% 2*round((1178-2230*min(fresult)/.75)/2)
m=log(b_f/a_fi)/log((a_fi-tott)/(b_f-tott));
m=2*(round(m/2))
a_f=(b_f/a_fi)^(1/m)
fprintf(fid, 'edge mesh "ax_mid" successive ratio1 %f intervals %f\n',1/a_f,m+2);

fprintf(fid, 'edge mesh "ax_mid1" size %f\n',a_fi);

len_ax=xyy(le-1)-xyy(ind+round(le/8));
tott=len_ax;
m=log(b_f/a_fi)/log((a_fi-tott)/(b_f-tott));
m=2*(round(m/2))
a_f=(b_f/a_fi)^(1/m)
fprintf(fid, 'edge mesh "ax_mid2" successive ratio1 %f intervals %f\n',a_f,m+2);

% b_f=grid_si-(.5*sten_le).*(stret-1)./stret;
% stret=(grid_si-a_fi)/(.5*sten_le)+1;
% m=round(log(grid_si/a_fi)/log(stret)+1)
sten_le=sum(sqrt((diff(fresult1([1:(ind-round(le/8))]))).^2+(diff(xyy1([1:(ind-
round(le/8))]))).^2));
tott=sten_le;
m=log(b_f/a_fi)/log((a_fi-tott)/(b_f-tott));
m=2*(round(m/2))
a_f=(b_f/a_fi)^(1/m)
fprintf(fid, 'edge mesh "sten" successive ratio1 %f intervals %f\n',1/a_f,m+2);
% fprintf(fid, 'edge mesh "ax_mid" successive ratio1 1 intervals 620\n');
% sten_le=sum(sqrt((diff(fresult1([(ind-
round(le/8)):(ind+round(le/8))]))).^2+(diff(xyy1([(ind-
round(le/8)):(ind+round(le/8))]))).^2));
fprintf(fid, 'edge mesh "sten1" size %f\n',a_fi);
mid_res=a_fi;
```

```
sten_le=sum(sqrt((diff(fresult1([(ind+round(le/8)):(le-
2)]))).^2+(diff(xyy1([(ind+round(le/8)):(le-2)]))).^2));
tott=sten_le;
m=log(b_f/a_fi)/log((a_fi-tott)/(b_f-tott));
m=2*(round(m/2))
a_f=(b_f/a_fi)^(1/m)
fprintf(fid, 'edge mesh "sten2" successive ratio1 %f intervals %f\n',a_f,m+2);

a_fi=b_f;
b_f=.75/8;
tott=xyy(1)--.75*15;
m=log(b_f/a_fi)/log((a_fi-tott)/(b_f-tott));
m=2*(round(m/2))
a_f=(b_f/a_fi)^(1/m)
% fprintf(fid, 'edge mesh "wall_prox" successive ratio1 1.005 intervals 250\n');
% fprintf(fid, 'edge mesh "ax_prox" successive ratio1 0.995 intervals 250\n');
% fprintf(fid, 'edge mesh "wall_dist" successive ratio1 0.995 intervals 250\n');
% fprintf(fid, 'edge mesh "ax_dist" successive ratio1 1.005 intervals 250\n');
fprintf(fid, 'edge mesh "wall_prox" successive ratio1 %f intervals %f\n',a_f,m+2);
fprintf(fid, 'edge mesh "ax_prox" successive ratio1 %f intervals %f\n',1/a_f,m+2);

tott=(3/2)*.75*15-xyy(le);
m=log(b_f/a_fi)/log((a_fi-tott)/(b_f-tott));
m=2*(round(m/2))
a_f=(b_f/a_fi)^(1/m)
fprintf(fid, 'edge mesh "wall_dist" successive ratio1 %f intervals %f\n',1/a_f,m+2);
fprintf(fid, 'edge mesh "ax_dist" successive ratio1 %f intervals %f\n',a_f,m+2);


a_fi=grid_si/4;
b_f=grid_si/2;
tott=.4*.75;
m=log(b_f/a_fi)/log((a_fi-tott)/(b_f-tott));
m=2*(round(m/2))
a_f=(b_f/a_fi)^(1/m)

fprintf(fid, 'blayer create first %f growth %f rows %f transition 1 trows 0
uniform\n',a_fi,a_f,m+2);
fprintf(fid, 'blayer attach "b_layer.1" face "prox" "dist" edge "wall_prox" "wall_dist"
add\n');

a_fi=mid_res/4
b_f=mid_res/2;
tott=.3*min(fresult);
m=log(b_f/a_fi)/log((a_fi-tott)/(b_f-tott));
m=2*(round(m/2))
```

```
a_f=(b_f/a_fi)^(1/m)
% a_first=b_f-B_f.*(a_f-1)./a_f
% rowss=round(log(b_f/a_first)./log(a_f))
% fprintf(fid, 'blayer create first 5 growth 1.1 total 30.5795 rows 15 transition 1 trows 0
aspectratio\n');
fprintf(fid, 'blayer create first %f growth %f rows %f transition 1 trows 0
uniform\n',a_fi,a_f,m+2);
% fprintf(fid, 'blayer create first %f total 80 rows 16 aspectlastratio\n',min(fresult)/300);
% fprintf(fid, 'blayer create first %f total 100 rows %f aspectlastratio\n',a_fi,m+2);


% fprintf(fid, 'blayer attach "b_layer.1" face "prox" "mid" "mid" "mid" "dist" edge
"wall_prox" "sten" "sten1" "sten2" "wall_dist" add\n');
fprintf(fid, 'blayer attach "b_layer.2" face "mid" "mid" "mid" edge "sten" "sten1" "sten2"
add\n');
% fprintf(fid, 'face mesh "mid" triangle size %f\n',.007);
% fprintf(fid, 'face mesh "prox" triangle size %f\n',.03);%0.2966*min(fresult)
% fprintf(fid, 'face mesh "dist" triangle size %f\n',.03);




b_f=.75/8;
tott=.75;
a_fi=.75/12;
m=log(b_f/a_fi)/log((a_fi-tott)/(b_f-tott));
m=2*(round(m/2))
a_f=(b_f/a_fi)^(1/m)

a_f=1.05;
B_f=tott;
m=(log(b_f)-log(a_f*(b_f-B_f)+B_f))/log(a_f);
m=2*(round(real(m)/2));
a_fi=b_f/(a_f^m)
a_f=(b_f/a_fi)^(1/m)

fprintf(fid, 'edge mesh "in" successive ratio1 %f intervals %f\n',a_f,2*m+2);
fprintf(fid, 'edge mesh "out" successive ratio1 %f intervals %f\n',1/a_f,2*m+2);


a_fi=b_f;
a_fi=grid_si*.5;
b_f=grid_si;
```

```
tott=.75;
% m=log(b_f/a_fi)/log((a_fi-tott)/(b_f-tott));
% m=2*(round(m/2))
% a_f=(b_f/a_fi)^(1/m)

a_f=1.05;
B_f=tott;
m=(log(b_f)-log(a_f*(b_f-B_f)+B_f))/log(a_f);
m=2*(round(real(m))/2);
a_fi=b_f/(a_f^m)
a_f=(b_f/a_fi)^(1/m)
a_f=1.02;
m=m-5;

fprintf(fid, 'edge mesh "prox_bord" successive ratio1 %f intervals %f\n',a_f,m+2);
fprintf(fid, 'edge mesh "dist_bord" successive ratio1 %f intervals %f\n',a_f,m+2);

a_f=1.15;
% b_f=(max(xyy)-min(xyy))./(2*round((1178-2230*min(fresult)/.75)/2));
b_f=min(fresult)/15;
B_f=.4*min(fresult);
m=(log(b_f)-log(a_f*(b_f-B_f)+B_f))/log(a_f);
m=2*(round(real(m))/2);
a_fi=b_f/(a_f^m)
a_f=(b_f/a_fi)^(1/m)
m_bound=m+1;




% fprintf(fid, 'face mesh "mid" pave \n');
fprintf(fid, 'face mesh "mid" triangle \n');
fprintf(fid, 'face mesh "prox" triangle \n');%0.2966*min(fresult)
fprintf(fid, 'face mesh "dist" triangle \n');%pave instead of triangle

fprintf(fid, 'physics create "vin" btype "VELOCITY_INLET" edge "in"\n');
fprintf(fid, 'physics create "pout" btype "PRESSURE_OUTLET" edge "out"\n');
% fprintf(fid, 'physics create "wall" btype "WALL" edge "wall_prox" "sten" "sten1"
"sten2" "wall_dist"\n');
fprintf(fid, 'physics create "wall" btype "WALL" edge "wall_prox" "sten" "sten1" "sten2"
"wall_dist"\n');
fprintf(fid, 'physics create "ax" btype "AXIS" edge "ax_prox" "ax_mid" "ax_mid1"
"ax_mid2" "ax_dist"\n');
fprintf(fid, 'solver select "FLUENT 5/6"\n');
fprintf(fid, ['export fluent5 "' y_nam 'sten%i.msh" nozval'],j+nam(1)-1);
```

```
fclose(fid);
system('C:\Fluent.Inc\ntbin\ntx86\gambit.exe -r2.4.6 -inp c:\SA2\sten.jou -new')
% system('C:\Program Files (x86)\StarNet\X-Win32 9.0\')
%
% %
fid = fopen('c:\SA2\stenfl.jou', 'wt');
fprintf(fid, ['file read-case ' y_nam 'sten%i.msh\n'],j+nam(1)-1);
fprintf(fid, 'grid scale .001 .001\n');
fprintf(fid, 'grid smooth-grid "skewness" 50 0.4\n');
fprintf(fid, 'define models axisymmetric yes\n');
fprintf(fid, 'define materials change-create air plasma y constant 1025 n n y constant
.0038 n n n n n n n \n');
fprintf(fid, 'define boundary-conditions fluid fluid y plasma n n y n n \n');
   fprintf(fid, 'define boundary-conditions velocity-inlet vin n n y y n %f \n',vell)%m/s
% fprintf(fid, 'define boundary-conditions velocity-inlet vin n n y y n %f \n',vel);%m/s
fprintf(fid, 'solve initialize initialize-flow \n');
fprintf(fid, 'solve monitors residual plot n \n');
fprintf(fid, 'solve monitors residual print y \n');
fprintf(fid, 'solve monitors residual convergence-criteria .0000001 .0000000000001
.001\n');
fprintf(fid, 'solve set p-v-coupling 24\n');
% fprintf(fid, 'solve set p-v-coupling 22\n');
% fprintf(fid, 'solve set p-v-controls 1 1 n\n');
% fprintf(fid, 'solve set discretization-scheme pressure 12\n');
% fprintf(fid, 'solve set discretization-scheme mom 1\n');
fprintf(fid, 'solve set discretization-scheme pressure 10\n');
fprintf(fid, 'solve set discretization-scheme mom 0\n');
% fprintf(fid, 'adapt refine-bndry-cells 4 () 10 0 1000000 y\n');
% fprintf(fid, 'adapt refine-bndry-cells 4 () %f 0 1000000 y\n',30);
% fprintf(fid, 'adapt refine-to-vol-change 2 1e-16 1000000 y\n');
% fprintf(fid, 'adapt refine-to-vol-change 2 1e-16 1000000 y\n');
% fprintf(fid, 'adapt refine-to-vol-change 2 1e-16 1000000 y\n');
% fprintf(fid, 'adapt refine-to-vol-change 2 1e-16 1000000 y\n');
fprintf(fid, 'file auto-save case-frequency 500\n');
fprintf(fid, 'file auto-save data-frequency 500\n');
fprintf(fid, 'file auto-save overwrite-existing-files y\n');
fprintf(fid, 'file auto-save max-files 1\n');
fprintf(fid, 'solve iterate 700\n');
fprintf(fid, ['file write ' y_nam 'sten%i.cas\n'],j+nam(1)-1);
fprintf(fid, ['file write ' y_nam 'sten%i.dat\n'],j+nam(1)-1);
fprintf(fid, ['file export ascii ' y_nam '%i- wall () y daxial-velocity-dy strain-rate-mag
pressure () n\n'],j+nam(1)-1);
fprintf(fid, 'exit y\n');

fclose(fid);
system('C:\Fluent.Inc\ntbin\win64\fluent.exe -r6.3.26 2ddp -i c:\SA2\stenfl.jou')
```

```
pause(120);
end
```

Appendix G: Refraction Correction (Chapter 3)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Using Snell's law to correct for refraction
%David Bark
%11/11/10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
a=sym('a');
b=sym('b');
g=sym('g');
d=sym('d');
y=sym('y');
% y3=sym('y3');
n1=sym('n1');
n2=sym('n2');
n3=sym('n3');
R=sym('R');
r=sym('r');

b=asin(n1*y/(n2*R));
g=solve((R*cos(a)-r*sin(pi/2-a+b-g))/cos(a-b)-(R*sin(a)-r*cos(pi/2-a+b-g))/sin(a-b),g);
% a=asin(y/R);
d=asin(n2*sin(g)/n3);
yprime=r*cos(pi/2-a+b-g)-r*cos(a-b+g)*tan(a-b+g-d);
% yprime=subs(yprime,[a n1 n3],[asin(y/R) 1 1.4]);
yprime=subs(yprime,[a n1],[asin(y/R) 1])

l=r*cos(a-b+g)/cos(a-b+g-d);
l=subs(l,[a n1 n3],[asin(y/R) 1 1.4]);

len=sym('len');
a=sym('a');
b=sym('b');
g=sym('g');
d=sym('d');
y=sym('y');
% y3=sym('y3');
n1=sym('n1');
n2=sym('n2');
n3=sym('n3');
R=sym('R');
r=sym('r');
```

```
g=solve((R*cos(a)-len*cos(a-b))/sin(pi/2-a+b-g)-(R*sin(a)-len*sin(a-b))/cos(pi/2-a+b-
g),g);
% a=asin(y/R);

d=asin(n2*sin(g)/n3);
rrr=len*cos(a-b+g-d)/cos(a-b+g);
rrr=subs(rrr,[a b],[asin(y/R) asin(n1*y/(n2*R))]);
yprime=subs(rrr,[n1 n3],[1 1.4]);

% yprime=subs(yprime,[R],[1500])

% moo=[];
% for i=.1:.05:.74
%     moo=[moo subs(yprime,[y],[i])];
% end
% i=[.1:.05:.74];
% figure
% plot(i,moo)



a=sym('a');
b=sym('b');
y=sym('y');
% y3=sym('y3');
n1=sym('n1');
n2=sym('n2');
R=sym('R');
r=sym('r');

b=asin(n1*y/(n2*R));
a=asin(y/R);

yprime=(y-R*cos(a)*tan(a-b));
yprime=subs(yprime,[n1],[1]);
yprime=subs(yprime,[R],[1500]);
boo=subs(yprime,y,1060);

boo=solve(750-boo,n2);
double(boo)

a=sym('a');
b=sym('b');
y=sym('y');
```

```
% y3=sym('y3');
n1=sym('n1');
n2=sym('n2');
R=sym('R');
r=sym('r');

b=asin(n1*y/(n2*R));
a=asin(y/R);

yprime=(y-R*cos(a)*tan(a-b));
yprime=subs(yprime,[n1 n2],[1 1.45])%dobule(boo)
```

Appendix H: Creating Single Plots for Each Studied Case (Chapter 3)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Creating plots for chapter 3
%David Bark
%11/11/10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all
clear all


n=['c04';'c23';'c24';'c26';'c29';'c30';'b77';'b74';'b82';'bc9'];
% n=['c04';'c23';'c24';'c26';'c29';'c30';'b77';'b74'; 'b82'; 'bc9'; 'b68'; 'd02'];
n=['c04';'c23';'c24';'c26';'c29';'c30'];
n=['c04';'c23';'c24';'c26';'c30'];
% n=['c04';'c26'];
expr.c04.file='C:\SA2\fix12_final_C4_22Jul200912_42.mat';
expr.c23.file='C:\SA2\fix7_final_C23_22Jul2009_10_24.mat';
expr.c24.file='C:\SA2\fix4_final_C24_26Aug2009_11_04.mat';
expr.c26.file='C:\SA2\fix5_final_C26_04Sep2009_12_19.mat';
expr.c29.file='C:\SA2\new_t_C29_28Sep2009_11_06.mat';
expr.c30.file='C:\SA2\fix8_final_C30_28Sep2009_10_44.mat';
expr.b77.file='C:\SA2\fix2_final_B77_29Jun2009_10_08.mat';
expr.b74.file='C:\SA2\fix2_final_B74_06Jul2009_09_48.mat';
expr.b82.file='C:\SA2\fix3_final_B82_15Jun2009_11_12.mat';
expr.bc9.file='C:\SA2\fix_alt_final_BC9_22Jul2009_09_55.mat';
expr.b68.file='C:\SA2\b68_17Mar2010_10_48.mat';
expr.d02.file='C:\SA2\d2_17Mar2010_15_12.mat';


expr.c04.y_nam='C:\SA2\C4_22Jul2009_12_42_\';
expr.c23.y_nam='C:\SA2\C23_22Jul2009_10_24_\';
expr.c24.y_nam='C:\SA2\C24_26Aug2009_11_04_\';
expr.c26.y_nam='C:\SA2\C26_04Sep2009_12_19_\';
expr.c29.y_nam='C:\SA2\C29_28Sep2009_11_06_\';
expr.c30.y_nam='C:\SA2\C30_28Sep2009_10_44_\';
expr.b77.y_nam='C:\SA2\B77_29Jun2009_10_08_\';
expr.b74.y_nam='C:\SA2\B74_06Jul2009_09_48_\';
expr.b82.y_nam='C:\SA2\B82_15Jun2009_11_12_\';
expr.bc9.y_nam='C:\SA2\BC9_22Jul2009_09_55_\';
expr.b68.y_nam='C:\SA2\b68_17Mar2010_10_48_\';
expr.d02.y_nam='C:\SA2\d2_17Mar2010_15_12_\';

expr.c04.k=[97 100:25:225 248 275:25:425 434 442 450 458 466 474];
```

```
expr.c04.k=[97 100:25:225 248 275:25:425 434 442 450 458];
expr.c04.k=[97 100:25:225 248 280 320 340:10:400 420:10:430 450:10:460];
expr.c04.k=[97 340 380 400 420 440 460];
expr.c04.k=[94 100 110 130:10:460];
expr.c04.k=[100:10:460];
expr.c23.k=[65 75:25:250 268 275 282 284 292 300 308 317 325 334 342 350];
expr.c23.k=[65 75:25:250 260:10:350];
expr.c23.k=[66 100:10:150 180 200:10:250 270:10:350];
expr.c23.k=[65 70:10:310 330 340 350];
% expr.c23.k=[65 280 350];
expr.c24.k=[35 50:25:200 210 212 218 220 225 230 232 238 244 250];
expr.c24.k=[35 40:10:230 242];
expr.c26.k=[4 50 75 100 125 150 163 175 180:10:250];
expr.c26.k=[4 50 75 100:10:190 200:10:240];
expr.c26.k=[3 10:10:240];
expr.c29.k=[28 50 75 100 125 150 175 200 212 225 238 242];
expr.c29.k=[28 120 180 200];
expr.c30.k=[56 100 110:5:135 150 200 225 250 275 284 292 300 308 316 325 333 342
350 361];
expr.c30.k=[56 60:10:350];
expr.c30.k=[55 60:10:230 250:10:350 359];
% expr.b77.k=[94 125:25:650];
expr.b77.k=[94 125:25:550 600 625 650];
expr.b74.k=[100:25:250 270:5:360 375:25:650];
expr.b74.k=[60:10:290 310:10:650];
expr.b82.k=[26 230:10:410];
expr.bc9.k=[56 60:10:650];
% expr.b68.k=[50:50:850];
expr.b68.k=[50:50:700];
expr.d02.k=[30 60 150:50:450 460];
expr.d02.k=[30 60 150:50:450 460];
cas=[5 6 7 8 9 10 4 1 2 3];
cas=[1 2 3 4 5 6 7 8 9 10];
normaliz= [.25 .42 .265 .514 .803 .54 0 0 0 0]*1e12
%
% [A B]=sort(cas);
% n=n(B,:);
% cas=cas(B);
% col=col(B);
% normaliz=normaliz(B);

% normaliz= [0 0 0 0 0 0];%flow rates for syringe pump expreriments
for i=[1:length(n)]
   eval(['expr.', n(i,:),'.normaliz=',num2str(normaliz(i)),';'])
   eval(['expr.', n(i,:),'.cas=',num2str(cas(i)),';'])
   eval(['expr.', n(i,:),'.col=',(col(i)),";'])
```

```
%     eval(['expr.', n(i,:),'.mark=',char((mark(i)),';'])
%     eval(['expr.', n(i,:),'.lin=',lin(i),';'])
end

thresho=.001; %degree of growth interpolation
phs1tim=5;
% n=['c04'];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
for iv=[1:length(n(:,1))]
    % iv=2;
    eval(['load(expr.',n(iv,:),'.file);'])
      x=x';
    xy={x,tim};
%     if (iv==5)||(iv==6)
%         x=x';
%     end
    eval(['expr.', n(iv,:),'.x=[',num2str(x'),'];'])
    eval(['expr.', n(iv,:),'.tim=[',num2str(tim'),'];'])
    eval(['k=[expr.', n(iv,:),'.k];'])
    eval(['y_nam=[expr.', n(iv,:),'.y_nam];'])
    eval(['normaliz=[expr.', n(iv,:),'.normaliz];'])

    %     H = fspecial('motion',20,45);
    Q=timq*(pi*.00075^2);
    H = fspecial('motion',20,0);
    im = imfilter(dat,H,'replicate');%filtering high freq from data noise
    rim = imfilter(rr,H,'replicate');%filtering high freq from data noise




    [a b]=max(im(:,length(im(1,:))));

%     figure
% %     plot(x.*ll,rim(:,length(tim)),'LineWidth',5,'LineStyle','-','Color','b')
%     fill([min(x.*ll) x'.*ll max(x.*ll)],[0 rim(:,length(tim))' 0],'r')
%     hold on
%     fill([min(x.*ll) x'.*ll max(x.*ll)],[1000 rim(:,length(tim))' 1000],'c')
%     fill([min(x.*ll) x'.*ll max(x.*ll)],[2000 rim(:,1)' 2000],'w')
% %     plot(x.*ll,rim(:,1),'LineWidth',5,'LineStyle','--','Color','k')
%     plot([x(b) x(b)].*ll,[0 750],'LineWidth',5,'LineStyle','-','Color','k')
% %     grid on
% xlim([-2000 2000])
%     set(gca,'FontSize',30);
%     set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
```

```
%     ylabel('Radius (microns)','FontSize',35);
%     xlabel('Axial Location (microns)','FontSize',35);
%     axis equal
% %     plot(tim,rr(j(2),:)/750,'LineWidth',5,'LineStyle','--','Color','k')
x2=ll.*x';
rim2=r_bot;
rim3=r_top;

% rim2=rr;
% rim3=rr;
if min(x2)>-2000
    x2=[-3000 x2];
    rim2=[rim2(1,:);rim2];
    rim3=[rim3(1,:);rim3];
end
if max(x2)<2000
    x2=[x2 3000];
    rim2=[rim2;rim2(length(rim2(:,1)),:)];
    rim3=[rim3;rim3(length(rim3(:,1)),:)];
end
x2=x2./750;
rim2=rim2./750;
rim3=rim3./750;

figure1=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
%     plot(x.*ll,rim(:,length(tim)),'LineWidth',5,'LineStyle','-','Color','b')
    fill([x2 fliplr(x2)],[-rim3(:,1)' fliplr(rim2(:,1)')],[.9 .9 .9])%'r'
    hold on

    aaa=double(tim>=300).*tim+(9e10)*double(tim<300);
    [aaa bbb]=min(aaa);
    fill([min(x2) x2 max(x2)],[1 rim2(:,bbb)' 1],[.0 .0 .0])
    fill([min(x2) x2 max(x2)],[-1 -rim3(:,bbb)' -1],[.0 .0 .0])

%     aaa=double(tim>=360).*tim+(9e10)*double(tim<360);
%     [aaa bbb]=min(aaa);
%     fill([min(x2) x2 max(x2.*ll)],[1 rim2(:,bbb)' 1],[.10 .10 .10])
%     fill([min(x2) x2 max(x2.*ll)],[-1 -rim3(:,bbb)' -1],[.10 .10 .10])

    aaa=double(tim>=240).*tim+(9e10)*double(tim<240);
    [aaa bbb]=min(aaa);
    fill([min(x2) x2 max(x2)],[1 rim2(:,bbb)' 1],[.20 .20 .20])
    fill([min(x2) x2 max(x2)],[-1 -rim3(:,bbb)' -1],[.20 .20 .20])

%     aaa=double(tim>=280).*tim+(9e10)*double(tim<280);
%     [aaa bbb]=min(aaa);
```

```
%     fill([min(x2) x2 max(x2)],[1 rim2(:,bbb)' 1],[.30 .30 .30])
%     fill([min(x2) x2 max(x2)],[-1 -rim3(:,bbb)' -1],[.30 .30 .30])

    aaa=double(tim>=180).*tim+(9e10)*double(tim<180);
    [aaa bbb]=min(aaa);
    fill([min(x2) x2 max(x2)],[1 rim2(:,bbb)' 1],[.40 .40 .40])
    fill([min(x2) x2 max(x2)],[-1 -rim3(:,bbb)' -1],[.40 .40 .40])
%     fill([min(x2) x2 max(x2)],[1 rim2(:,bbb)' 1],[.30 .30 .30])
%     fill([min(x2) x2 max(x2)],[-1 -rim3(:,bbb)' -1],[.30 .30 .30])

%     aaa=double(tim>=200).*tim+(9e10)*double(tim<200);
%     [aaa bbb]=min(aaa);
%     fill([min(x2) x2 max(x2)],[1 rim2(:,bbb)' 1],[.50 .50 .50])
%     fill([min(x2) x2 max(x2)],[-1 -rim3(:,bbb)' -1],[.50 .50 .50])

    aaa=double(tim>=120).*tim+(9e10)*double(tim<120);
    [aaa bbb]=min(aaa);
    fill([min(x2) x2 max(x2)],[1 rim2(:,bbb)' 1],[.60 .60 .60])
    fill([min(x2) x2 max(x2)],[-1 -rim3(:,bbb)' -1],[.60 .60 .60])
%     fill([min(x2) x2 max(x2)],[1 rim2(:,bbb)' 1],[.40 .40 .40])
%     fill([min(x2) x2 max(x2)],[-1 -rim3(:,bbb)' -1],[.40 .40 .40])

%     aaa=double(tim>=120).*tim+(9e10)*double(tim<120);
%     [aaa bbb]=min(aaa);
%     fill([min(x2) x2 max(x2)],[1 rim2(:,bbb)' 1],[.70 .70 .70])
%     fill([min(x2) x2 max(x2)],[-1 -rim3(:,bbb)' -1],[.70 .70 .70])

    aaa=double(tim>=60).*tim+(9e10)*double(tim<60);
    [aaa bbb]=min(aaa);
    fill([min(x2) x2 max(x2)],[1 rim2(:,bbb)' 1],[.80 .80 .80])
    fill([min(x2) x2 max(x2)],[-1 -rim3(:,bbb)' -1],[.80 .80 .80])
%     fill([min(x2) x2 max(x2)],[1 rim2(:,bbb)' 1],[.50 .50 .50])
%     fill([min(x2) x2 max(x2)],[-1 -rim3(:,bbb)' -1],[.50 .50 .50])
%
%     aaa=double(tim>=40).*tim+(9e10)*double(tim<40);
%     [aaa bbb]=min(aaa);
%     fill([min(x2) x2 max(x2)],[1 rim2(:,bbb)' 1],[.90 .90 .90])
%     fill([min(x2) x2 max(x2)],[-1 -rim3(:,bbb)' -1],[.90 .90 .90])



    fill([min(x2) x2 max(x2)],[2000 rim2(:,1)' 2000],'w')
    fill([min(x2) x2 max(x2)],[-2000 -rim3(:,1)' -2000],'w')
%     plot(x,rim(:,1),'LineWidth',5,'LineStyle','--','Color','k')

    plot([x(b) x(b)].*ll./750,[-750 750]./750,'LineWidth',5,'LineStyle','--','Color','k')
```

```
    set(gca,'FontSize',30);
    set(gca,'position',[0.1    0.2000    0.7157    0.7250]);
    set(gca,'TickDir','out');
    ylabel('Normalized Radius (r/r_0)','FontSize',35);
    xlabel('Normalized Axial Location (z/r_0)','FontSize',35);
    axis equal
    xlim([-3 3])
    ylim([-1 1])
    [po]=get(gca,'Position');
%     annotation('arrow',[po(1)+po(3)/3 po(1)+po(3)/3+po(3)/3],[po(4)/2+po(2)
po(4)/2+po(2)],'HeadWidth',50,'LineWidth',5);
    title(['Case:',num2str(eval(['expr.', n(iv,:),'.cas']))])
%     colormap([.9;.8;.7;.6;.5;.4;.3;.2;.1;.0]*[1 1 1])
%     labels = {'1.5','3.0','4.5','6.0','7.5','9.0','10.5','12.0','13.5','15.0'};
    colormap([.8;.6;.4;.2;.0]*[1 1 1])
    labels = {'1','2','3','4','5','6'};
    lcolorbar(labels,'fontsize',20);
    annotation2 =
annotation(figure1,'textbox','LineStyle','none','rotation',90,'FontSize',30,'Position',[0.8703
-0.5929 0.1 0.8547],'FitHeightToText','off','String',{'Thrombus Boundary at
Time','(minutes) Shown in Gray'});
% %     text([0],[0],cellstr({'Thrombus','Boundary','at Time (minutes)','Shown
in','Gray'}),'rotation',90,'FontSize',30)
%     im=mean(im([(b-5):1:(b+5)],:));
%     rim=mean(rim([(b-5):1:(b+5)],:));
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%
%
%
%
%     %%%%%graph%%%%%%%%%%%%%%%
%     %     plot(im(700,:))
%     %     hold on
%     %     plot(dat(700,:),'r')
%     %
%     %     plot(im(650,:))
%     %     plot(im(750,:))
%     %     plot(im(600,:))
%     %
%     %     plot(dat(650,:),'r')
%     %     plot(dat(750,:),'r')
%     %     plot(dat(600,:),'r')
%     %     figure
%     %     contourf(im)
%     %     figure
```

```
%    %     contourf(dat)
%
%    %%%%%%%%%%Evaluate Shear in comp domain%%%%%%%%%%
%    shrinf=[];
%    shrhist=[];
%    press=[];
%    for j=k
%       fid = fopen([y_nam,num2str(j),'-'],'r');
%       shrdat = textscan(fid,'%f %f %f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
%       shrdat=cell2mat(shrdat);
%       shrx=1000*(shrdat(:,2));
%       shry=1000*(shrdat(:,3));
%       pres=(shrdat(:,4));
%       pres=diff(pres)./diff(sqrt(shrx.^2+shry.^2));
%       pres=[pres(1);pres];
%       shrm=(shrdat(:,5));
%       shrm=-shrm.*sign((shrdat(:,6)));
%       fclose(fid);
%       %         fit1=malowess(ll.*x, shrinf, 'Robust', 'on')';
%       %         shrinf=fit1';
%       sr=csaps(shrx,shrm,.9995,ll.*x/1000);
%
%       %         shrinf=[shrinf fnval(ll.*x/1000,csaps(shrx,shrm))];%x in mm
%       shrinf=[shrinf sr];%x in mm
%       press=[press csaps(shrx,pres,.9995,ll.*x/1000)];
%       if j==k(1)
%          shrst=zeros(size(sr));
%          shrhist=[shrhist shrst];
%       else
%          shrst=shrst+(tim(j-nam(1)+1)-tim(j-1-nam(1)+1)).*sr.^2.3;
%          shrhist=[shrhist shrst];
%       end
%    end
%    shrinf=mean(shrinf([(b-5):1:(b+5)],:));
%    s=(eval(['expr.', n(iv,:),'.normaliz'])).*4./pi./(rim.^3);
%    shrinf=fnval(tim,csaps(tim(k-nam(1)+1),shrinf));
%    %%%%%%%%%%%%%%%%%%%%dep rate%%%%%%
%    dppp2=[];
%    for i=1
%       doo=(-60.*(.8/8).*(diff(rim(i,:))./diff(tim')));
%       for j=1:(length(tim)-1)
%          if (abs(doo(j))>100000)
%             if j==1
%                doo(j)=doo(j+1);
%             else
%                doo(j)=doo(j-1);
```

```
%             end
%           end
%       end
%       dppp2=[dppp2;doo];
%   end
%   dppp2=[dppp2 dppp2(:,length(dppp2(1,:)))];
%   H = fspecial('motion',60,0);
%   dppp3 = imfilter(dppp2,H,'replicate');
%
%   % %%other method
%   % dppp=[];
%   % depr=[];
%   % fresult =  csaps(xy,-60.*(.8/8).*rim,thresho,xy);
%   % for i=1:length(x)
%   %     fresult2=csaps(tim,fresult(i,:),thresho);
%   %     depr=[depr;fnval(tim,fresult2)'];
%   %     fresult2=fnder(fresult2);
%   %     fresult2=fnval(tim,fresult2);
%   %     dppp=[dppp;fresult2'];
%   % end
%   % figure
%   % contourf (dppp)
%   %
%
%   %%%%Calculate time initation%%%%%%
%
%
%   init=[];
%   rinit=[];
%   shrinit=[];
%   xinit=[];
%   % ti=[];
%   tj=[];
%   QQ=[];
%   for j=1:length(x)
%       for i=2:length(tim)
%           if((dat(j,i)>phs1tim)&&(dat(j,i-1)<phs1tim))
%               %           ti=[ti i];
%               tj=[tj j];
%               init=[init tim(i)];
%               xinit=[xinit x(j)];
%               rinit=[rinit rr(j,i)];
%               %           QQ=[QQ Q(i)*60000000];
%               break
%           end
%       end
```

```
%     end
%
%     %%%%%%%%%%%%%%%%%%%%convert to shear data%%%%%%%%%%
%     separationint =  csaps(xinit,init,.05,x);
%
separationint=(x>max(xinit)).*100000000+(x<min(xinit)).*100000000+separationint;
%     grwthterm=[];
%     shrterm=[];
%     xterm=[];
%     sinit=[];
%     sxinit=[];
%     srinit=[];
%     sgradd=[];
%     tjj=[];
% %     for jj=1:length(k)
%         shrterm=[shrterm shrinf'];
%         grwthterm=[grwthterm dppp3];
%         xterm=[xterm ll.*x(b)];%ll.*x(j)/1000];
% %     end
%     %     shrterm=shrterm(grwthterm~=0);
%     %     grwthterm=grwthterm(grwthterm~=0);
%     %     sgradd=sgradd(grwthterm~=0);
%     %     xterm=xterm(grwthterm~=0);
%     %     sum(grwthterm==0)
%
%
%     eval(['expr.', n(iv,:),'.shear=[',num2str(shrterm),'];'])
%     eval(['expr.', n(iv,:),'.grw=[',num2str(grwthterm),'];'])
%     eval(['expr.', n(iv,:),'.rim=[',num2str(rim),'];'])
%     eval(['expr.', n(iv,:),'.im=[',num2str(im),'];'])
%     eval(['expr.', n(iv,:),'.xgrw=[',num2str(xterm),'];'])
  [a b]=min(rim(:,1));
  eval(['expr.', n(iv,:),'.r=[',num2str(rim(b,:)),'];'])
  eval(['expr.', n(iv,:),'.tim=[',num2str(tim'),'];'])


end
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
plot(expr.c04.tim,expr.c04.r(1)-expr.c04.r,'Linestyle','-','color',[0 0 0],'linewidth',4);
hold on
plot(expr.c23.tim,expr.c23.r(1)-expr.c23.r,'Linestyle','--','color',[.2 .2 .2],'linewidth',4);
plot(expr.c24.tim,expr.c24.r(1)-expr.c24.r,'Linestyle','-.','color',[.4 .4 .4],'linewidth',4);
plot(expr.c26.tim,expr.c26.r(1)-expr.c26.r,':','color',[.6 .6 .6],'linewidth',4);
plot(expr.c30.tim,expr.c30.r(1)-expr.c30.r,'Linestyle','-','color',[.8 .8 .8],'linewidth',4);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
```

```
legend('Case 1','Case 2','Case 3','Case 4','Case 5')
grid on
xlabel('Time (s)','FontSize',35);
ylabel('Thrombus Thickness (\mum)','FontSize',35);
%% Create annotation
% annotation1 = annotation(...
%   figure1,'line',...
%   [0.3536 0.7745],[0.842 0.1792],...
%   'LineWidth',6,...
%   'LineStyle','--');
%
% %% Create annotation
% annotation2 = annotation(...
%   figure1,'line',...
%   [0.5781 0.4849],[0.64 0.64],...
%   'LineWidth',6,...
%   'LineStyle','--');
%
% %% Create annotation
% annotation3 = annotation(...
%   figure1,'line',...
%   [0.5766 0.5766],[0.4916 0.6421],...
%   'LineWidth',6,...
%   'LineStyle','--');
%  text('String','dy/dt','Position',[.5 .5],'FontSize',30)


%
% %% Create textbox
% annotation5 = annotation(...
%   figure1,'textbox',...
%   'Position',[0.5625 0.6705 0.08281 0.06947],...
%   'BackgroundColor',[1 1 1],...
%   'LineStyle','none',...
%   'FitHeightToText','off',...
%   'FontSize',30,...
%   'String',{'\frac{a}{x}'},...
%   'Interpreter','latex');
% text('Interpreter','latex','String','\frac{\partialr}{\partialt}','Position',[.5
.5],'FontSize',30)

% text('Interpreter','latex','String','$$\int_0^x\!\int_y dF(u,v)$$','Position',[.5
.5],'FontSize',30)


%
```

```
% %%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%end%%%%
%%%%%%%%%
%
% % n=['c04';'c23';'c24';'c26';'c29';'c30'];
%
%
%
%
%
% figure
%
plot(expr.c04.shear,expr.c04.grw,'Linestyle','none','Marker','o','MarkerEdgeColor','k','Mar
kerFaceColor','k','MarkerSize',5);
% hold on
%
plot(expr.c23.shear,expr.c23.grw,'Linestyle','none','Marker','o','MarkerEdgeColor','r','Mar
kerFaceColor','r','MarkerSize',5);
%
plot(expr.c24.shear,expr.c24.grw,'Linestyle','none','Marker','o','MarkerEdgeColor','b','Mar
kerFaceColor','b','MarkerSize',5);
%
plot(expr.c26.shear,expr.c26.grw,'Linestyle','none','Marker','o','MarkerEdgeColor','g','Mar
kerFaceColor','g','MarkerSize',5);
%
plot(expr.c29.shear,expr.c29.grw,'Linestyle','none','Marker','o','MarkerEdgeColor','m','Ma
rkerFaceColor','m','MarkerSize',5);
%
plot(expr.c30.shear,expr.c30.grw,'Linestyle','none','Marker','o','MarkerEdgeColor','y','Mar
kerFaceColor','y','MarkerSize',5);
%
%
%
plot(expr.b77.shear,expr.b77.grw,'Linestyle','none','Marker','s','MarkerEdgeColor','k','Mar
kerFaceColor','k','MarkerSize',5);
%
plot(expr.b74.shear,expr.b74.grw,'Linestyle','none','Marker','s','MarkerEdgeColor','r','Mar
kerFaceColor','r','MarkerSize',5);
%
plot(expr.b82.shear,expr.b82.grw,'Linestyle','none','Marker','s','MarkerEdgeColor','b','Mar
kerFaceColor','b','MarkerSize',5);
%
plot(expr.bc9.shear,expr.bc9.grw,'Linestyle','none','Marker','s','MarkerEdgeColor','g','Mar
kerFaceColor','g','MarkerSize',5);
%
```

```
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% grid on
% xlabel('Shear Rate (1/s)','FontSize',35);
% ylabel('Phase II Platelets/microns^2-min','FontSize',35);
%
%
%
% figure
% plot(expr.c04.tim,expr.c04.im,'Linestyle','-','Linewidth',3,'Color','k');
% hold on
% plot(expr.c23.tim,expr.c23.im,'Linestyle','-','Linewidth',3,'Color','r');
% plot(expr.c24.tim,expr.c24.im,'Linestyle','-','Linewidth',3,'Color','b');
% plot(expr.c26.tim,expr.c26.im,'Linestyle','-','Linewidth',3,'Color','g');
% plot(expr.c29.tim,expr.c29.im,'Linestyle','-','Linewidth',3,'Color','m');
% plot(expr.c30.tim,expr.c30.im,'Linestyle','-','Linewidth',3,'Color','y');
%
%
% plot(expr.b77.tim,expr.b77.im,'Linestyle','--','Linewidth',3,'Color','k');
% plot(expr.b74.tim,expr.b74.im,'Linestyle','--','Linewidth',3,'Color','r');
% plot(expr.b82.tim,expr.b82.im,'Linestyle','--','Linewidth',3,'Color','b');
% plot(expr.bc9.tim,expr.bc9.im,'Linestyle','--','Linewidth',3,'Color','g');
%
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% grid on
% xlabel('Time (s)','FontSize',35);
% ylabel('Growth (microns)','FontSize',35);
%
% figure
% plot(expr.b74.tim,expr.b74.rim,'Linestyle','--','Linewidth',3,'Color','r');
% hold on
% plot(expr.b82.tim,expr.b82.rim,'Linestyle','--','Linewidth',3,'Color','b');
% plot(expr.bc9.tim,expr.bc9.rim,'Linestyle','--','Linewidth',3,'Color','g');
% plot(expr.b77.tim,expr.b77.rim,'Linestyle','--','Linewidth',3,'Color','k');
%
% plot(expr.c04.tim,expr.c04.rim,'Linestyle','-','Linewidth',3,'Color','k');
% plot(expr.c23.tim,expr.c23.rim,'Linestyle','-','Linewidth',3,'Color','r');
% plot(expr.c24.tim,expr.c24.rim,'Linestyle','-','Linewidth',3,'Color','b');
% plot(expr.c26.tim,expr.c26.rim,'Linestyle','-','Linewidth',3,'Color','g');
% plot(expr.c29.tim,expr.c29.rim,'Linestyle','-','Linewidth',3,'Color','m');
% plot(expr.c30.tim,expr.c30.rim,'Linestyle','-','Linewidth',3,'Color','y');
%
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% grid on
```

```
% xlabel('Time (s)','FontSize',35);
% ylabel('Radius (microns)','FontSize',35);
% legend(['Case:',num2str(eval(['expr.', n(1,:),'.cas']))],['Case:',num2str(eval(['expr.',
n(2,:),'.cas']))],['Case:',num2str(eval(['expr.', n(3,:),'.cas']))],['Case:',num2str(eval(['expr.',
n(4,:),'.cas']))],['Case:',num2str(eval(['expr.', n(5,:),'.cas']))],['Case:',num2str(eval(['expr.',
n(6,:),'.cas']))],['Case:',num2str(eval(['expr.', n(7,:),'.cas']))],['Case:',num2str(eval(['expr.',
n(8,:),'.cas']))],['Case:',num2str(eval(['expr.', n(9,:),'.cas']))],['Case:',num2str(eval(['expr.',
n(10,:),'.cas']))])
%
% figure
% plot(expr.b74.tim,expr.b74.shear,'Linestyle','--','Linewidth',3,'Color','r');
% hold on
% plot(expr.b82.tim,expr.b82.shear,'Linestyle','--','Linewidth',3,'Color','b');
% plot(expr.bc9.tim,expr.bc9.shear,'Linestyle','--','Linewidth',3,'Color','g');
% plot(expr.b77.tim,expr.b77.shear,'Linestyle','--','Linewidth',3,'Color','k');
%
% plot(expr.c04.tim,expr.c04.shear,'Linestyle','-','Linewidth',3,'Color','k');
% plot(expr.c23.tim,expr.c23.shear,'Linestyle','-','Linewidth',3,'Color','r');
% plot(expr.c24.tim,expr.c24.shear,'Linestyle','-','Linewidth',3,'Color','b');
% plot(expr.c26.tim,expr.c26.shear,'Linestyle','-','Linewidth',3,'Color','g');
% plot(expr.c29.tim,expr.c29.shear,'Linestyle','-','Linewidth',3,'Color','m');
% plot(expr.c30.tim,expr.c30.shear,'Linestyle','-','Linewidth',3,'Color','y');
%
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% grid on
% xlabel('Time (s)','FontSize',35);
% ylabel('Shear Rate (1/s)','FontSize',35);
% legend(['Case:',num2str(eval(['expr.', n(1,:),'.cas']))],['Case:',num2str(eval(['expr.',
n(2,:),'.cas']))],['Case:',num2str(eval(['expr.', n(3,:),'.cas']))],['Case:',num2str(eval(['expr.',
n(4,:),'.cas']))],['Case:',num2str(eval(['expr.', n(5,:),'.cas']))],['Case:',num2str(eval(['expr.',
n(6,:),'.cas']))],['Case:',num2str(eval(['expr.', n(7,:),'.cas']))],['Case:',num2str(eval(['expr.',
n(8,:),'.cas']))],['Case:',num2str(eval(['expr.', n(9,:),'.cas']))],['Case:',num2str(eval(['expr.',
n(10,:),'.cas']))])
%
%
% shry=[expr.c04.shear expr.c23.shear expr.c24.shear expr.c26.shear expr.c29.shear
expr.c30.shear];
% grwy=[expr.c04.grw expr.c23.grw expr.c24.grw expr.c26.grw expr.c29.grw
expr.c30.grw];
% xxy=[expr.c04.xgrw expr.c23.xgrw expr.c24.xgrw expr.c26.xgrw expr.c29.xgrw
expr.c30.xgrw];
%
%
% grwy2=grwy(shry>3000);
% shry2=shry(shry>3000);
```

```
% % shry=[expr.c23.shear expr.c24.shear expr.c26.shear expr.c29.shear expr.c30.shear];
% % grwy=[expr.c23.grw expr.c24.grw expr.c26.grw expr.c29.grw expr.c30.grw];
% % xxy=[expr.c23.xgrw expr.c24.xgrw expr.c26.xgrw expr.c29.xgrw expr.c30.xgrw];
% % sgrady=sign([expr.c23.sgrad expr.c24.sgrad expr.c26.sgrad expr.c29.sgrad
expr.c30.sgrad]).*[expr.c23.shear expr.c24.shear expr.c26.shear expr.c29.shear
expr.c30.shear];
%
% shrbin=[20000 100000000];
% [nn,xbins]=histc(shry,shrbin);%xbins is the bin location
% figure
% hist(grwy(xbins==1),50);
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% grid on
% xlabel('Phase II Platelets/microns^2-min','FontSize',35);
% ylabel('# Instances','FontSize',35);
%
% % grown=grwy(xbins==1);
% % shown=shry(xbins==1);
% % std(grown)
% % mean(grown)
% % xbin=[min(x).*ll -2000 -1000:200:400 max(x).*ll];%x bins in microns
%
% shrbin=[-inf -10000 -3000 -1000 -500 0:100:3000 3500:500:9500
10000:20000:200000 inf];
%
% sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
% if (sum(shry<-10000)>0)
%     sb=[mean(shry(shry<shrbin(2))) sb];%x bin values
% else
%     sb=[-20000 sb];
% end
% if (sum(shry>shrbin(length(shrbin)-1))>0)
%     sb=[sb mean(shry(shry>shrbin(length(shrbin)-1)))];%x bin values
% else
%     sb=[sb shrbin(length(shrbin)-1)+10000];
% end
%
%
% [nn,xbins]=histc(shry,shrbin);%xbins is the bin location
% gbin=[];
% gbinn=[];
% sb2=[];
% for i=[1:length(sb)]
%     if (sum(grwy(xbins==i))~=0)
%         gbin=[gbin (mean(grwy(xbins==i)))];%averaged growth
```

```
%         gbinn=[gbinn std((grwy(xbins==i)))];%averaged growth
%         sb2=[sb2 sb(i)];
%     end
% end
% sbb=shry(shry>=5000);
% gbb=grwy(shry>=5000);
% figure
% hh=errorbar(sb2,gbin,gbinn,'o');
% xlim([-1000 10000]);
% errorbar_tick(hh)
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% grid on
% xlabel('Shear Rate (1/s)','FontSize',35);
% ylabel('Phase II (Platelets/microns^2-min)','FontSize',35);
% %%%%transort prediction%%%
% gam=[100:100:100000];
% D=.004.*((7e-9).*gam+1.6e-9);%cm^2/s
% D=((2.5e-9).*gam+1.6e-9);%cm^2/s
% D=1.6e-9
% z=.00025;%platelet diameter cm
% cinf=50.*2.75e8;%platelet/cm^3
% j=(60.*1e-8).*cinf./(1.8575./(D.*(gam./(D.*z)).^(1/3)));%convert cm to microns and
seconds to minutes
% hold on
% plot(gam,j,'r')
% %binn
%
% % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %
% %
% % for i=1:(length(k)-1)
% % figure
% % [AX,H1,H2]=plotyy(x.*ll./750,dppp3(:,(k(i)-
nam(1)+1)),x.*ll./750,shrinf(:,i));%(k(i)-nam(1)+1))
% % xlim(AX(2),[-2 2]);
% % xlim(AX(1),[-2 2]);
% % ylim(AX(2),[0 40000]);
% % ylim(AX(1),[-2 10]);
% % set(AX(1),'FontSize',30);
% % set(AX(1),'position',[0.1893    0.2000    0.65    0.7250]);
% % set(AX(2),'FontSize',30);
% % set(AX(2),'position',[0.1893    0.2000    0.65    0.7250]);
% % set(AX(1),'XColor','k','YColor','b')
% % set(AX(2),'XColor','k','YColor','k')
% % set(get(AX(2),'Ylabel'),'String','Shear Rate(s^-^1)','FontSize',35)
```

```
% % % set(AX(1),'ylim',[0 1]);
% % % set(AX(1),'YTick',[0:.2:1]);
% % % set(AX(2),'YTick',[-10000:10000:40000]);
% % % ylim(AX(2),[min(roundn(shrinf(:,i),4))-10000
max(roundn(shrinf(:,i),4))+10000]);
% % set(get(AX(1),'Ylabel'),'String','Thrombus Flux (platelets/micron^2-
min)','FontSize',35)
% % set(get(AX(1),'Xlabel'),'String','Axial Location/R_0','FontSize',35)
% % set(get(AX(2),'Xlabel'),'String','Axial Location/R_0','FontSize',35)
% % set(H1,'LineWidth',5,'LineStyle','-','Color','b')
% % set(H2,'LineWidth',5,'LineStyle','--','Color','k')
% % %
set(H2,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',15,'LineWidt
h',5,'LineStyle','-','Color','k')
% % % ylabel('Time (s)','FontSize',20);
% % % set(get(AX(1),'Title'),'String','%f',k(i),'FontSize',35)
% %
% % grid on;%
% % tit=tim(k(i)-nam(1)+1);
% % tit=num2str(tit);
% % tit=['Time: ' tit ' s'];
% % set(get(gca,'Title'),'String',tit,'FontSize',35)
% % end
```

Appendix I: Creating Plots (Chapter 3)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Creating plots for chapter 3
%David Bark
%11/11/10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%end%%%%%
%%%%%%%%%%%
close all
% n=['c04';'c23';'c24';'c26';'c30';'bc9'];
% n=['c04';'c23';'c24';'c26';'c29';'c30';'b77';'b74';'b82';'bc9'];
% n=['c04';'c23';'c24';'c26';'c30'];
% n=['c04';'c23';'c24';'c26';'c30';'bc9'];
% n=['bc9'];
n=['b77';'b74';'b82';'bc9'];
% n='c30';
n=['c04';'c23';'c24';'c26';'c30';'b77';'b74';'b82';'bc9'];
symb={'o','s','^','>','<','v','.','*','+'};
% symb={'','','','','<','v','.','*','+'};
symbl={'-';'--';'-.';':';'-';'--';'-.';':';'-'};

alph= fittype('60*(1e-8)*(2.75e8)*c*k/(1+1.8575*k/((a*x+1.6e-
9)*(x/(.00025*(a*x+1.6e-9)))^(1/3)))');
alph2= fittype('60*(1e-8)*(2.75e8)*c/(1/(k*x^n)+1.8575/((a*x+1.6e-
9)*(x/(.00025*(a*x+1.6e-9)))^(1/3)))');
% alph= fittype('60*(1e-8)*(2.75e8)*k/(1+1.8575*k/(x*(a^(2/3))))'); %x=shear/x^(1/3)
act= fittype('a*(x^b)+c');
cinf=2.75e8;%platelet/cm^3
z=.00025;%platelet diameter cm
grwy=[];
% if 1==0
%     for i=1:length(n(:,1))
%         grwy=[grwy (eval(['expr.', n(i,:),'.grw'])./(eval(['expr.', n(i,:),'.normaliz']))];
%     end
% else
%     for i=1:length(n(:,1))
%         grwy=[grwy (eval(['expr.', n(i,:),'.grw']))];
%     end
% end
shry=[];
xxy=[];
sgrady=[];
```

```
sgradyy=[];
tiy=[];
rimy=[];
iy=[];
pressy=[];
srinity=[];
inity=[];
xinity=[];
grwdat=[];
shryg=[];
xxyg=[];
sgradyg=[];
sgradyyg=[];
tiyg=[];
rimyg=[];
iyg=[];
pressyg=[];
srinityg=[];
inityg=[];
xinityg=[];
grwdatg=[];
grwyg=[];

gvel=[];
gvelg=[];
nli=[300 250 260 325 325];
rvals=[];
rvals2=[];

shrbin=logspace(1,5,100);
sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
if (sum(shry<-10000)>0)
   sb=[mean(shry(shry<shrbin(2))) sb];%x bin values
else
   sb=[-20000 sb];
end
if (sum(shry>shrbin(length(shrbin)-1))>0)
   sb=[sb mean(shry(shry>shrbin(length(shrbin)-1)))];%x bin values
else
   sb=[sb shrbin(length(shrbin)-1)+10000];
end
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
% figure
hold on
chg=10;
```

```
xbin=[-750-chg -750+chg -375-chg -375+chg -chg chg 375-chg 375+chg 750-chg
750+chg];
xbin_lab=[-750 -375 0 375 750];
rvals_x=[];
rvals_t=[];
chg=10;
tbin=[60-chg 60+chg 120-chg 120+chg 180-chg 180+chg 240-chg 240+chg 300-chg
300+chg];

for i=1:length(n(:,1))

   if i<=5
      if i==1
         slim=5000;
      elseif i==2
         slim=8000;
      elseif i==3
         slim=10000;
      elseif i==4
         slim=5000;
      else
         slim=4000;
      end

      sr=(eval(['expr.', n(i,:),'.shear']));
      gr=(eval(['expr.', n(i,:),'.grw']));
      xr=(eval(['expr.', n(i,:),'.xgrw']));
      tr=(eval(['expr.', n(i,:),'.ti']));
      sgr=(eval(['expr.', n(i,:),'.sgrad']));
      [cr1 cp]=corrcoef([gr((sr<slim)&(xr<0)&(sr>0))'
(abs(sr((sr<slim)&(xr<0)&(sr>0))))']);
      %       [cr2 cp]=corrcoef([gr((sr>10000)&(sr<60000)&(xr<0))'
1./(sr((sr>10000)&(sr<60000)&(xr<0)).^.07)']);
      [cr2 cp]=corrcoef([gr((sr>slim)&(sr<100000)&(xr<0))'
(sr((sr>slim)&(sr<100000)&(xr<0)))']);
      rvals=[rvals; cr1(1,2) cr2(1,2)];
      %       xr=xr+1000;
      %       [cr1 cp]=corrcoef([gr((sr<slim))' (abs(sr((sr<slim))).^3)'.*(xr(sr<slim)')]);
      %       [cr1 cp]=corrcoef([gr((sr<slim))' (abs(sr((sr<slim))).^3)'.*(xr(sr<slim)')]);
      %       rvals2=[rvals2; cr1(1,2)];
      % gr(gr<=0)=1e-10;
      % stepwise([log10(abs(tr((sr<5000)&(xr<0))))'
log10(abs(sr((sr<5000)&(xr<0))))'],log10(abs(gr((sr<5000)&(xr<0)))+1e-10)')

      trr=tr((sr<5000)&(sr>0));
      xrr=xr((sr<5000)&(sr>0));
```

```
srr=sr((sr<5000)&(sr>0));
grr=gr((sr<5000)&(sr>0));
[nn,xbins]=histc(xrr,xbin);%xbins is the bin location
gbin=[];
for ivv=[1 3 5 7 9]
   if (sum(grr(xbins==ivv))~=0)
      cor=corrcoef([grr(xbins==ivv)' srr(xbins==ivv)']);%averaged growth
      gbin=[gbin cor(1,2)];
   else
      gbin=[gbin 0];
   end
end
rvals_x=[rvals_x; gbin];

grr=grr(xrr<0);
srr=srr(xrr<0);
trr=trr(xrr<0);
[nn,xbins]=histc(trr,tbin);%xbins is the bin location
gbin=[];
for ivv=[1 3 5 7 9]
   if (sum(grr(xbins==ivv))~=0)
      cor=corrcoef([grr(xbins==ivv)' srr(xbins==ivv)']);%averaged growth
      gbin=[gbin cor(1,2)];
   else
      gbin=[gbin 0];
   end
end
rvals_t=[rvals_t; gbin];



sr=sr(xr<0);
gr=gr(xr<0);
[nn,xbins]=histc(sr,shrbin);%xbins is the bin location
gbin=[];
gbinn=[];
sb2=[];
for ivv=[1:length(sb)]
   if (sum(gr(xbins==ivv))~=0)
      gbin=[gbin (mean(gr(xbins==ivv)))];%averaged growth
      gbinn=[gbinn std((gr(xbins==ivv)))];%./sum(xbins==i)];%averaged growth
      sb2=[sb2 sb(ivv)];
   end
end
```

```
%
hh=semilogx(sb2,gbin,'Marker',symb(i),'MarkerEdgeColor','k','MarkerFaceColor','none','
MarkerSize',10,'LineStyle','none');
      semilogx(sb2,(8.8./.8).*gbin,'linestyle',char(symbl(i)),'color','k', 'linewidth',
3,'Marker',char(symb(i)),'MarkerEdgeColor','k','MarkerFaceColor','none','MarkerSize',4);
      % hh=semilogx(sb2,gbin+gbinn,'--');



      grwy=[grwy (eval(['expr.', n(i,:),'.grw']))];
      shry=[shry (eval(['expr.', n(i,:),'.shear']))];
      rimy=[rimy (eval(['expr.', n(i,:),'.rim']))];
      xxy=[xxy (eval(['expr.', n(i,:),'.xgrw']))];
      tiy=[tiy (eval(['expr.', n(i,:),'.ti']))];

      iy=[iy i.*ones(1,length(eval(['expr.', n(i,:),'.shear'])))];
      sgrady=[sgrady sign(eval(['expr.', n(i,:),'.sgrad'])).*(eval(['expr.', n(i,:),'.shear']))];
      sgradyy=[sgradyy (eval(['expr.', n(i,:),'.sgrad']))];
      %     sgradyy=[sgradyy (eval(['expr.', n(i,:),'.sgrad'])./(eval(['expr.', n(i,:),'.shear']))];

      pressy=[pressy (eval(['expr.', n(i,:),'.p']))];
      srinity=[srinity (eval(['expr.', n(i,:),'.srinit']))];
      inity=[inity (eval(['expr.', n(i,:),'.init']))];
      xinity=[xinity (eval(['expr.', n(i,:),'.xinit']))];
      grwdat=[grwdat (eval(['expr.', n(i,:),'.grwdat']))];
      gvel=[gvel (eval(['expr.', n(i,:),'.vel']))];
   else
      grwyg=[grwyg (eval(['expr.', n(i,:),'.grw']))];
      shryg=[shryg (eval(['expr.', n(i,:),'.shear']))];
      rimyg=[rimyg (eval(['expr.', n(i,:),'.rim']))];
      xxyg=[xxyg (eval(['expr.', n(i,:),'.xgrw']))];
      tiyg=[tiyg (eval(['expr.', n(i,:),'.ti']))];

      iyg=[iyg i.*ones(1,length(eval(['expr.', n(i,:),'.shear'])))];
      sgradyg=[sgradyg sign(eval(['expr.', n(i,:),'.sgrad'])).*(eval(['expr.', n(i,:),'.shear']))];
      sgradyyg=[sgradyyg (eval(['expr.', n(i,:),'.sgrad']))];
      %     sgradyyg=[sgradyyg (eval(['expr.', n(i,:),'.sgrad'])./(eval(['expr.',
n(i,:),'.shear']))];

      pressyg=[pressyg (eval(['expr.', n(i,:),'.p']))];
      srinityg=[srinityg (eval(['expr.', n(i,:),'.srinit']))];
      inityg=[inityg (eval(['expr.', n(i,:),'.init']))];
      xinityg=[xinityg (eval(['expr.', n(i,:),'.xinit']))];
      grwdatg=[grwdatg (eval(['expr.', n(i,:),'.grwdat']))];
      gvelg=[gvelg (eval(['expr.', n(i,:),'.vel']))];
   end
```

```
end
legend('Case 1','Case 2','Case 3','Case 4','Case 5')

set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
grid on
xlabel('Shear Rate (1/s)','FontSize',35);
ylabel('Thrombus Growth Rate (\mum^3/\mum^2-min)','FontSize',35);
xlim([100 100000]);
ylim([0 35]);
set(gca,'xscale','log');

rvals_x
rvals_t

shryt=[shry shryg];
grwyt=[grwy grwyg];
tiyt=[tiy tiyg];
xxyt=[xxy xxyg];
grwdatt=[grwdat grwdatg];
rimyt=[rimy rimyg];

% shry=shry(grwdat<40);
% grwy=grwy(grwdat<40);
% tiy=tiy(grwdat<40);
% xxy=xxy(grwdat<40);
% rimy=rimy(grwdat<40);
% sgrady=sgrady(grwdat<40);
% sgradyy=sgradyy(grwdat<40);
% pressy=pressy(grwdat<40);
% gvel=gvel(grwdat<40);
% grwdat=grwdat(grwdat<40);

rvals
corrl=[grwy' grwdat' (abs(shry))' rimy' xxy' tiy' sgrady' pressy' gvel'.*750.*750./(rimy'.^2)
1025.*2.*gvel'.*750.*750.*(1e-6)./(rimy')./.0038 abs(shry')./(rimy)'];
corrl=corrl((shry>0)&(shry<5000));
[cr1 cp]=corrcoef([corrl]);
cr1(cp>=.05)=0;

cr1

% grwy=grwy+min(grwy)+1e-10;

%
```

```
% fittl=[grwdat((shry<5000)&(xxy<0)&(shry>0))'
(abs(shry((shry<5000)&(xxy<0)&(shry>0))))' rimy((shry<5000)&(xxy<0)&(shry>0))' -
xxy((shry<5000)&(xxy<0)&(shry>0))' tiy((shry<5000)&(xxy<0)&(shry>0))' ];
% fittl2=grwy((shry<5000)&(xxy<0)&(shry>0))';
% fittl2=fittl2-min(fittl2)+1e-10;
% stepwise(log10(fittl),log10(fittl2));

% stepwise([grwdat((shry<5000))' (abs(shry((shry<5000))))' rimy((shry<5000))'
xxy((shry<5000))' tiy((shry<5000))' sgrady((shry<5000))'
1025.*2.*gvel((shry<5000))'.*750.*750.*(1e-6)./(rimy((shry<5000))')./.0038
],grwy((shry<5000))')
%
% stepwise([grwdat((shry<5000)&(xxy<0))' (abs(shry((shry<5000)&(xxy<0))))'
rimy((shry<5000)&(xxy<0))' xxy((shry<5000)&(xxy<0))' tiy((shry<5000)&(xxy<0))'
1025.*2.*gvel((shry<5000)&(xxy<0))'.*750.*750.*(1e-
6)./(rimy((shry<5000)&(xxy<0))')./.0038 ],grwy((shry<5000)&(xxy<0))')

figure
plot(tiy((shry<5000)&(xxy<0)),grwy((shry<5000)&(xxy<0)),'.')
figure
plot(tiyg((shryg<5000)&(xxyg<0)),grwyg((shryg<5000)&(xxyg<0)),'.')

% stepwise([grwdatt((shryt<5000)&(xxyt<0))' (abs(shryt((shryt<5000)&(xxyt<0))))'
rimyt((shryt<5000)&(xxyt<0))' xxyt((shryt<5000)&(xxyt<0))'
tiyt((shryt<5000)&(xxyt<0))' ],grwyt((shryt<5000)&(xxyt<0))')



figure




for i=1:length(n)
   figure
   tootle=eval(['expr.', n(i,:),'.hyd']);
   %     tootle=eval(['expr.', n(i,:),'.hyd'])+mean(tootle(1:20))-min(eval(['expr.',
n(i,:),'.hyd']));
   %     tootle=tootle/tootle(1);
   eval(['plot(expr.',
n(i,:),'.tim_shr,[',num2str(tootle),'],"Linestyle","none","Marker","o","MarkerEdgeColor",[',
num2str([1 1 1]./(length(n)+2-
i)),'],"MarkerFaceColor","none","MarkerSize",10);'])%[',num2str([1 1 1]./(length(n)+1-
i)),']
```

```
    toot=eval(['expr.', n(i,:),'.tim_shr']);%_shr
    %       [p2,S2] = polyfit((toot),tootle,4);
    %     [pop2,del2] = polyval(p2,(toot),S2);
    %     plot(toot,tootle,'+',toot,pop2,'g-',toot,pop2+2*del2,'r:',toot,pop2-2*del2,'r:'), grid
on


    hold on
    tootle=eval(['expr.', n(i,:),'.hyde']);
    %     tootle=tootle/tootle(3);
    %     tootle=tootle-mean(tootle(1:20))+min(eval(['expr.', n(i,:),'.hyd']));
    %     tootle=tootle/tootle(1);


    eval(['plot(expr.',
n(i,:),'.tim,[',num2str(tootle),'],"Linestyle","none","Marker","x","MarkerEdgeColor",[',nu
m2str([1 1 1]./(length(n)+2-
i)),'],"MarkerFaceColor","none","MarkerSize",10);'])%[',num2str([1 1 1]./(length(n)+1-
i)),']




    set(gca,'FontSize',30);
    set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
    grid on
    xlabel('Time (s)','FontSize',35);
    %     ylabel('\DeltaP/(0.5{\rho}V^2)','FontSize',35);
    ylabel('\DeltaP','FontSize',35);
    title(['Case:',num2str(([n(i,:)]))])
    %     title(['Case:',num2str(eval(['expr.', n(i,:),'.cas']))])
    %     hold off
end

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
shrbin=[10:10:500];
% shrbin=[0:100:5000]
grwy2=grwy;
grwy=grwyg;
sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
if (sum(tiyg<-10000)>0)
    sb=[mean(tiyg(tiyg<shrbin(2))) sb];%x bin values
else
    sb=[-20000 sb];
end
```

```
if (sum(tiyg>shrbin(length(shrbin)-1))>0)
    sb=[sb mean(tiyg(tiyg>shrbin(length(shrbin)-1)))];%x bin values
else
    sb=[sb shrbin(length(shrbin)-1)+10000];
end
grwy=8.8.*grwy./.8

[nn,xbins]=histc(tiyg,shrbin);%xbins is the bin location
gbin=[];
gbinn=[];
sb2=[];
for i=[1:length(sb)]
    if (sum(grwy(xbins==i))~=0)
        gbin=[gbin (mean(grwy(xbins==i)))];%averaged growth
        gbinn=[gbinn std((grwy(xbins==i)))];%./sum(xbins==i)];%averaged growth
        sb2=[sb2 sb(i)];
    end
end
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
hh=semilogx(sb2,gbin,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','Marke
rSize',10,'LineStyle','none');
xlim([10 100000]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
grid on
xlabel('Time (s)','FontSize',35);
ylabel('Thrombus Growth Rate (\mum^3/\mum^2-min)','FontSize',35);
grwy=grwy2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
exprs=[212 1690 3380 265 630 2100 100 650 2600 420 2600 10500 32000];
exprd=[0.067794 0.38826 1.5936 0.01998 0.04998 0.36 0.036 0.108 0.396 0.1575 0.63
1.035 0.45];
%%%%%%%%%%%%%%%%%%phase
I%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%pressure comp%%%%%%
shrbin=[0:100:1000 1000:500:6000 6000:1000:9000 10000:10000:200000 inf];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%best%%%%%%%%%%%%%
%%%
shrbin=[0:200:800 1000:500:6000 6000:1000:9000 10000:10000:200000 inf];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
shrbin=logspace(1,5,100);
% shrbin=[0:100:5000]
grwy2=grwy;
```

```
sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
if (sum(shry<-10000)>0)
   sb=[mean(shry(shry<shrbin(2))) sb];%x bin values
else
   sb=[-20000 sb];
end
if (sum(shry>shrbin(length(shrbin)-1))>0)
   sb=[sb mean(shry(shry>shrbin(length(shrbin)-1)))];%x bin values
else
   sb=[sb shrbin(length(shrbin)-1)+10000];
end
grwy=8.8.*grwy./.8

[nn,xbins]=histc(shry,shrbin);%xbins is the bin location
gbin=[];
gbinn=[];
sb2=[];
for i=[1:length(sb)]
   if (sum(grwy(xbins==i))~=0)
      gbin=[gbin (mean(grwy(xbins==i)))];%averaged growth
      gbinn=[gbinn std((grwy(xbins==i)))];%./sum(xbins==i)];%averaged growth
      sb2=[sb2 sb(i)];
   end
end
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
hh=semilogx(sb2,gbin,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','Marke
rSize',10,'LineStyle','none');
xlim([10 100000]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
grid on
xlabel('Shear Rate (1/s)','FontSize',35);
ylabel('Thrombus Growth Rate (\mum^3/\mum^2-min)','FontSize',35);


figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
st=100*(1-(450./sb2(sb2>=450)).^(1./3));
st=(2.52e-4).*sb2(sb2>=450)+47.2;
hh=plot(st,1500.*(((450./sb2(sb2>=450)).^(1./3))./gbin(sb2>=450),'Marker','o','MarkerE
dgeColor','k','MarkerFaceColor','none','MarkerSize',10,'LineStyle','none');
hh=plot(st,1500.*(1-
st./100)./gbin(sb2>=450),'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','Ma
rkerSize',10,'LineStyle','none');

% xlim([10 100000]);
set(gca,'FontSize',30);
```

```
set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
grid on
xlabel('% Stenosis by Diameter','FontSize',35);
ylabel('Occlusion Time (min)','FontSize',35);
% xlim([0 80])
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
hh=plot(100*(1-
(450./sb2(sb2>=450)).^(1./3)),gbin(sb2>=450),'Marker','o','MarkerEdgeColor','k','Marker
FaceColor','none','MarkerSize',10,'LineStyle','none');
% xlim([10 100000]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
grid on
xlabel('% Stenosis by Diameter','FontSize',35);
ylabel('Thrombus Growth Rate (\mum^3/\mum^2-min)','FontSize',35);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
hh=semilogx(sb2,gbin,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','Marke
rSize',10,'LineStyle','none');
hold on
ftype = fittype('a1.*x.^b1');
[fd gof1]=fit(sb((sb>0)&(sb<4000))',gbin((sb>0)&(sb<4000))',ftype,'StartPoint',[.1
.33],'Lower',[0 0],'Robust','on');
gof1
a1 =      fd.a1
b1 =      fd.b1
plot(sb((sb>0)&(sb<10000)),a1.*(sb((sb>0)&(sb<10000)).^(b1)),'linestyle','-.','color','k',
'linewidth', 5);
ftype = fittype('a1.*x.^b1+c1');
[fd
gof2]=fit(sb((sb>10000)&(sb<60000))',gbin((sb>10000)&(sb<60000))',ftype,'StartPoint',
[-20 .093 70.6]);
gof2
a1 =      fd.a1
b1 =      fd.b1
c1 =      fd.c1
plot(sb((sb>2000)&(sb<100000)),a1.*sb((sb>2000)&(sb<100000)).^b1+c1,'linestyle',':','c
olor','k', 'linewidth', 5);
plot(sb2,gbinn+gbin,'LineStyle','--','color','k', 'linewidth', 3);
plot(sb2,gbin-gbinn,'LineStyle','--','color','k', 'linewidth', 3);
xlim([10 100000]);
legend('Mean','Curve Fit I','Curve Fit II','Standard Deviation');
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
grid on
```

```
xlabel('Shear Rate (1/s)','FontSize',35);
ylabel('Thrombus Growth Rate (\mum^3/\mum^2-min)','FontSize',35);
ylim([-10 35]);
xxybin=[-inf 0];
[nn,xxybins]=histc(xxy,xxybin);%xbins is the bin location
xbins(xxybins~=1)=-5;
grwy3=grwy;
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
hold on
clr=jet(size(n,1));
for ii=1:size(n(:,1))
   [nn,xbins]=histc((eval(['expr.', n(ii,:),'.shear'])),shrbin);%xbins is the bin location
   gbin=[];
   gbinn=[];
   sb2=[];
   grwy=(eval(['expr.', n(ii,:),'.grw']));
   for i=[1:length(sb)]
      if (sum(grwy(xbins==i))~=0)
         gbin=[gbin (mean(grwy(xbins==i)))];%averaged growth
         gbinn=[gbinn std((grwy(xbins==i)))];%./sum(xbins==i)];%averaged growth
         sb2=[sb2 sb(i)];
      end
   end
   %    figure
   if (length(sb2)>0)
      if ii<=5

semilogx(sb2,gbin,'Linestyle','none','Marker','o','MarkerEdgeColor',clr(ii,:),'MarkerFaceC
olor',clr(ii,:),'MarkerSize',10);
      else

semilogx(sb2,gbin,'Linestyle','none','Marker','s','MarkerEdgeColor',clr(ii,:),'MarkerFaceC
olor',clr(ii,:),'MarkerSize',10);
      end
   end
   xlim([10 100000]);
end
legend('c04','c23','c24','c26','c30','b77','b74','b82','bc9');
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
grid on
xlabel('Shear Rate (1/s)','FontSize',35);
ylabel('Phase I Platelets/microns^2-min','FontSize',35);
grwy=grwy3;
```

```
grwy=grwy2;
sb3=sb;
%%%%% location%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Re=1025.*2.*gvel'.*750.*750.*(1e-6)./(rimy')./.0038;
shrbin=[1:.1:50];
sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
if (sum(Re<-10000)>0)
   sb=[mean(Re(Re<shrbin(2))) sb];%x bin values
else
   sb=[-20000 sb];
end
if (sum(Re>shrbin(length(shrbin)-1))>0)
   sb=[sb mean(Re(Re>shrbin(length(shrbin)-1)))];%x bin values
else
   sb=[sb shrbin(length(shrbin)-1)+10000];
end


[nn,xbins]=histc(Re,shrbin);%xbins is the bin location
gbin=[];
gbinn=[];
sb2=[];
for i=[1:length(sb)]
   if (sum(grwy(xbins==i))~=0)
      gbin=[gbin (mean(grwy(xbins==i)))];%averaged growth
      gbinn=[gbinn std((grwy(xbins==i)))];%./sum(xbins==i)];%averaged growth
      sb2=[sb2 sb(i)];
   end
end
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
hh=plot(sb2,gbin,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','MarkerSize
',10,'LineStyle','none');
hold on
plot(sb2,gbinn+gbin,'LineStyle','--','color','k', 'linewidth', 3);
plot(sb2,gbin-gbinn,'LineStyle','--','color','k', 'linewidth', 3);
xlim([1 30]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
grid on
xlabel('Re','FontSize',35);
ylabel('Thrombus Growth Rate (\mum^3/\mum^2-min)','FontSize',35);


shry2=shry;
grwy2=grwy;
```

```
grwy=grwy((Re<8)&(Re<20));
Re=shry((Re<8)&(Re<20));
grwy=grwy(Re>0)
Re=Re(Re>0);


shrbin=logspace(1,6,100);
sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
if (sum(Re<-10000)>0)
   sb=[mean(Re(Re<shrbin(2))) sb];%x bin values
else
   sb=[-20000 sb];
end
if (sum(Re>shrbin(length(shrbin)-1))>0)
   sb=[sb mean(Re(Re>shrbin(length(shrbin)-1)))];%x bin values
else
   sb=[sb shrbin(length(shrbin)-1)+10000];
end


[nn,xbins]=histc(Re,shrbin);%xbins is the bin location
gbin=[];
gbinn=[];
sb2=[];
for i=[1:length(sb)]
   if (sum(grwy(xbins==i))~=0)
      gbin=[gbin (mean(grwy(xbins==i)))];%averaged growth
      gbinn=[gbinn std((grwy(xbins==i)))];%./sum(xbins==i)];%averaged growth
      sb2=[sb2 sb(i)];
   end
end
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
hh=semilogx(sb2,gbin,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','Marke
rSize',10,'LineStyle','none');
hold on
plot(sb2,gbinn+gbin,'LineStyle','--','color','k', 'linewidth', 3);
plot(sb2,gbin-gbinn,'LineStyle','--','color','k', 'linewidth', 3);
xlim([1 10^6]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
grid on
xlabel('Shear Rate','FontSize',35);
ylabel('Thrombus Growth Rate (\mum^3/\mum^2-min)','FontSize',35);

[cr cp]=corrcoef([grwy' 1./Re'])
```

```
shry=shry2;
grwy=grwy2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
grwy2=grwy;
shry2=shry;
shry=shry(xxy>=0);
grwy=grwy(xxy>=0);
shrbin=logspace(1,6,100);
sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
if (sum(shry<-10000)>0)
   sb=[mean(shry(shry<shrbin(2))) sb];%x bin values
else
   sb=[-20000 sb];
end
if (sum(shry>shrbin(length(shrbin)-1))>0)
   sb=[sb mean(shry(shry>shrbin(length(shrbin)-1)))];%x bin values
else
   sb=[sb shrbin(length(shrbin)-1)+10000];
end
% grwy=8.8.*grwy./.8

[nn,xbins]=histc(shry,shrbin);%xbins is the bin location
gbin=[];
gbinn=[];
sb2=[];
for i=[1:length(sb)]
   if (sum(grwy(xbins==i))~=0)
      gbin=[gbin (mean(grwy(xbins==i)))];%averaged growth
      gbinn=[gbinn std((grwy(xbins==i)))];%./sum(xbins==i)];%averaged growth
      sb2=[sb2 sb(i)];
   end
end
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
hh=semilogx(sb2,gbin,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','Marke
rSize',10,'LineStyle','none');
hold on
plot(sb2,gbinn+gbin,'LineStyle','--','color','k', 'linewidth', 3);
plot(sb2,gbin-gbinn,'LineStyle','--','color','k', 'linewidth', 3);


grwy=grwy2;
shry=shry2;
shry=shry(xxy<0);
grwy=grwy(xxy<0);
shrbin=logspace(1,5,100);
```

```
sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
if (sum(shry<-10000)>0)
   sb=[mean(shry(shry<shrbin(2))) sb];%x bin values
else
   sb=[-20000 sb];
end
if (sum(shry>shrbin(length(shrbin)-1))>0)
   sb=[sb mean(shry(shry>shrbin(length(shrbin)-1)))];%x bin values
else
   sb=[sb shrbin(length(shrbin)-1)+10000];
end
% grwy=8.8.*grwy./.8

[nn,xbins]=histc(shry,shrbin);%xbins is the bin location
gbin=[];
gbinn=[];
sb2=[];
for i=[1:length(sb)]
   if (sum(grwy(xbins==i))~=0)
      gbin=[gbin (mean(grwy(xbins==i)))];%averaged growth
      gbinn=[gbinn std((grwy(xbins==i)))];%./sum(xbins==i)];%averaged growth
      sb2=[sb2 sb(i)];
   end
end

hh=semilogx(sb2,gbin,'Marker','o','MarkerEdgeColor','g','MarkerFaceColor','none','Marke
rSize',10,'LineStyle','none');
corrl=[gbin(sb2<7000)' 1./sb2(sb2<7000)'];
[cr cp]=corrcoef([corrl]);

hold on
plot(sb2,gbinn+gbin,'LineStyle','--','color','g', 'linewidth', 3);
plot(sb2,gbin-gbinn,'LineStyle','--','color','g', 'linewidth', 3);

plot([100 800 1500],[.0215 .0615 .10075],'Marker','h','MarkerEdgeColor',[.2 .2
.2],'MarkerFaceColor',[.2 .2 .2],'MarkerSize',10,'LineStyle','none');% Aleveduou 93
plot([212 1690 3380],[0.067794 0.38826 1.5936],'Marker','p','MarkerEdgeColor',[.4 .4
.4],'MarkerFaceColor',[.4 .4 .4],'MarkerSize',15,'LineStyle','none');%Badimon 86
plot([420 2600 10500 32000],[.25 1.09375 1.78125
.78125],'Marker','<','MarkerEdgeColor',[.2 .2 .2],'MarkerFaceColor',[.2 .2
.2],'MarkerSize',15,'LineStyle','none');%barstad_94
plot([420 2600 10500 32000],[.1875 .625 1.5 .625],'Marker','>','MarkerEdgeColor',[.2 .2
.2],'MarkerFaceColor',[.2 .2 .2],'MarkerSize',15,'LineStyle','none');%barstad_96
plot([11500 10000 24200 32700 42100 42200 40900],[1.2525 0.825 1.41 1.8 2.31 1.7775
2.7],'Marker','^','MarkerEdgeColor',[.2 .2 .2],'MarkerFaceColor',[.2 .2
.2],'MarkerSize',15,'LineStyle','none');%Flannery
```

```
% plot([500],[.0125],'Marker','v','MarkerEdgeColor',[.4 .4 .4],'MarkerFaceColor',[.4 .4
.4],'MarkerSize',10,'LineStyle','none');%Hubbel 86
plot([265 630 2100],[0.01998 0.04998 0.36],'Marker','d','MarkerEdgeColor',[.6 .6
.6],'MarkerFaceColor',[.6 .6 .6],'MarkerSize',15,'LineStyle','none');%Markou 93a
plot([265 630 900 2120 2400 7500],[0.1155 0.298 0.8255 1.103 1.2295
1.6855],'Marker','s','MarkerEdgeColor',[.6 .6 .6],'MarkerFaceColor',[.4 .4
.4],'MarkerSize',15,'LineStyle','none');%Markou 93b
plot([100 650 2600],[0.036 0.108 0.396],'Marker','x','MarkerEdgeColor',[.3 .3
.3],'MarkerFaceColor',[.3 .3 .3],'MarkerSize',15,'LineStyle','none');%Sakariassen 90

xlim([10 100000]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
grid on
xlabel('Shear Rate (1/s)','FontSize',35);
ylabel('Thrombus Growth Rate (\mum^3/\mum^2-min)','FontSize',35);
grwy=grwy2;
shry=shry2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%Phs1%%%%%%%%%%%



figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
plot(srinity(xinity<0),inity(xinity<0),'Linestyle','none','Marker','o','MarkerEdgeColor','k','
MarkerFaceColor','k','MarkerSize',10);
hold on

plot(srinity(xinity>=0),inity(xinity>=0),'Linestyle','none','Marker','s','MarkerEdgeColor','
k','MarkerFaceColor','k','MarkerSize',10);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
grid on
xlabel('Shear Rate (1/s)','FontSize',35);
ylabel('Lag Period (s)','FontSize',35);


figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
plot(srinity,inity,'Linestyle','none','Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k'
,'MarkerSize',10);
hold on

inity1=inity(srinity>=0)';
srinity1=srinity(srinity>=0)';
[rvalilin p]=corrcoef([srinity1,inity1])
[rvaliexp p]=corrcoef([(srinity1),log(inity1)])
[rvalipow p]=corrcoef([log10(srinity1),log10(inity1)])
```

```
[rvalirecip p]=corrcoef([1./srinity1,inity1])
srinity_=srinity1;
inity_=inity1;
srinity1=log10(srinity1);
inity1=log10(inity1);
b1=(length(srinity1).*(sum(srinity1.*inity1))-
sum(srinity1).*(sum(inity1)))./(length(srinity1).*sum(srinity1.^2)-sum(srinity1).^2)
a1=(sum(inity1)-b1.*sum(srinity1))./length(srinity1)
srinity1=srinity_;
inity1=inity_;
% ftype = fittype('a1.*exp(x.*b1)+c1');
%
% % [fd gof2]=fit(srinity1,inity1,ftype,'StartPoint',[8000 -.0063 740],'robust','on');
[fit1,gof1,out1] =fit(srinity1,inity1,'power1','robust','on');
% gof2
a1 =      fit1.a
b1 =      fit1.b
% c1 =      fd.c1
srr=[0:10:10000];
plot(srr,a1.*srr.^(b1),'linestyle','--','color','k', 'linewidth', 5);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
grid on
xlabel('Shear Rate (1/s)','FontSize',35);
ylabel('Lag Period (s)','FontSize',35);
xlim([0 6000])
legend('Lag Period','Curve Fit');


% figure
%
plot(xinity,(inity),'Linestyle','none','Marker','o','MarkerEdgeColor','k','MarkerFaceColor','
k','MarkerSize',10);
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% grid on
% xlabel('Axial Location','FontSize',35);
% ylabel('Lag Period (s)','FontSize',35);

shrbin=logspace(1,4,15);
sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
if (sum(srinity<-10000)>0)
   sb=[mean(srinity(srinity<shrbin(2))) sb];%x bin values
else
   sb=[-20000 sb];
end
```

```
if (sum(srinity>shrbin(length(shrbin)-1))>0)
   sb=[sb mean(srinity(srinity>shrbin(length(shrbin)-1)))];%x bin values
else
   sb=[sb shrbin(length(shrbin)-1)+10000];
end
[nn,xbins]=histc(srinity,shrbin);%xbins is the bin location
gbin=[];
gbinn=[];
sb2=[];
for i=[1:length(sb)]
   if (sum(inity(xbins==i))~=0)
      gbin=[gbin (mean(inity(xbins==i)))];%averaged growth
      gbinn=[gbinn std((inity(xbins==i)))];%./sum(xbins==i)];%averaged growth
      sb2=[sb2 sb(i)];
   end
end

figure
loglog(gbin,.038*sb2,'Linestyle','none','Marker','o','MarkerEdgeColor','k','MarkerFaceCol
or','k','MarkerSize',10);
hold on
% loglog(gbin,1000./(gbin.^(.452)),'LineWidth',5,'LineStyle','--','Color','k');
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
grid on
xlabel('Shear Rate (1/s)','FontSize',35);
ylabel('Lag Time (s)','FontSize',35);
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%Phs1%%%%%%%%%%%
%
%
% [nn,xbins]=histc(srinity,shrbin);%xbins is the bin location
%
% for i=[1:length(sb)]
%    if (sum(xbins==i)~=0)
%       inity(xbins==i)=inity(xbins==i)-(phs1tim-1)/(gbin(i)/(60.*.8./8.8));
%    end
% end
% grwinit=(60.*.8./8.8)./inity;
%
% figure
% hold on
% xbinn=[-750 -250 0 250 750];
% [nn,xbinns]=histc(xinity,xbinn);%xbins is the bin location
% clr=jet(length(xbinn)-1);
% for ii=1:(length(xbinn)-1)
%    sb1=srinity(ii==xbinns);
```

```
%     gb=grwinit(ii==xbinns);
%     if (length(sb1)>0)
%
plot(sb1,gb,'Linestyle','none','Marker','o','MarkerEdgeColor',clr(ii,:),'MarkerFaceColor',cl
r(ii,:),'MarkerSize',10);
%     end
%     alpha(.05);
% end
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% grid on
% xlabel('Shear Rate (1/s)','FontSize',35);
% ylabel('Phase I Platelets/microns^2-min','FontSize',35);

% %%%%%%%%%%%%%%%%%%phase
I%%%%%%%%%%%%%%%%%%%%%%%%%%
% grwinit=grwinit((xinity>=-1000)&(xinity<=1000));
% srinity=srinity((xinity>=-1000)&(xinity<=1000));
%
% plot(srinity,grwinit,'.')
% % shrbin=logspace(1,4,10);
% shrbin=logspace(1,4,15);
% sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
% if (sum(shry<-10000)>0)
%     sb=[mean(shry(shry<shrbin(2))) sb];%x bin values
% else
%     sb=[-20000 sb];
% end
% if (sum(shry>shrbin(length(shrbin)-1))>0)
%     sb=[sb mean(shry(shry>shrbin(length(shrbin)-1)))];%x bin values
% else
%     sb=[sb shrbin(length(shrbin)-1)+10000];
% end
% [nn,xbins]=histc(srinity,shrbin);%xbins is the bin location
% gbin=[];
% gbinn=[];
% sb2=[];
% for i=[1:length(sb)]
%     if (sum(grwinit(xbins==i))~=0)
%         gbin=[gbin (mean(grwinit(xbins==i)))];%averaged growth
%         gbinn=[gbinn std((grwinit(xbins==i)))];%./sum(xbins==i)];%averaged growth
%         sb2=[sb2 sb(i)];
%     end
% end
%
%
```

```
% gbin_ph1=gbin;
% gbinn_ph1=gbinn;
% sb_ph1=sb2;
% % gbin_ph1=grwinit;
% % sb_ph1=srinity;
% figure
%
hh=semilogx(sb2,gbin,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSi
ze',10,'LineStyle','none');
% hold on
%
% plot([500],[.0125],'Marker','*','MarkerEdgeColor',[.4 .4 .4],'MarkerFaceColor',[.4 .4
.4],'MarkerSize',10,'LineStyle','none');%Hubbel 86
% plot([100 800 1500],[.0215 .0615 .10075],'Marker','x','MarkerEdgeColor',[.2 .2
.2],'MarkerFaceColor',[.2 .2 .2],'MarkerSize',10,'LineStyle','none');% Aleveduou 93
% plot([420 2600],[.03056 .03565],'Marker','<','MarkerEdgeColor',[.2 .2
.2],'MarkerFaceColor',[.2 .2 .2],'MarkerSize',10,'LineStyle','none');%barstad_96
% % plot([11500 10000 24200 32700 42100 42200 40900],[1.2525 0.825 1.41 1.8 2.31
1.7775 2.7],'Marker','>','MarkerEdgeColor',[.2 .2 .2],'MarkerFaceColor',[.2 .2
.2],'MarkerSize',10,'LineStyle','none');%Flannery
% % plot([265 630 2100],[0.01998 0.04998 0.36],'Marker','+','MarkerEdgeColor',[.6 .6
.6],'MarkerFaceColor',[.6 .6 .6],'MarkerSize',10,'LineStyle','none');%Markou 93a
% % plot([265 630 900 2120 2400 7500],[0.1155 0.298 0.8255 1.103 1.2295
1.6855],'Marker','^','MarkerEdgeColor',[.6 .6 .6],'MarkerFaceColor',[.4 .4
.4],'MarkerSize',10,'LineStyle','none');%Markou 93b
% % plot([100 650 2600],[0.036 0.108 0.396],'Marker','v','MarkerEdgeColor',[.8 .8
.8],'MarkerFaceColor',[.8 .8 .8],'MarkerSize',10,'LineStyle','none');%Sakariassen 90
% plot(sb2,gbinn+gbin,'LineStyle','--');
% plot(sb2,gbin-gbinn,'LineStyle','--');
%
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% grid on
% xlabel('Shear Rate (1/s)','FontSize',35);
% ylabel('Phase I Platelets/microns^2-min','FontSize',35);
%


% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%compare phase 1 and 2
% %
% xxbins=[];
% xxbinss=[];
% sb=[];
% shrbin=[-1000 1000 -900 1100 -800 1200 -700 1300 -600 1400 -500 1500 -400 1600 -
300 1700 -200 1800 -100 1900 0 2000 1000 3000 4000 6000 9000 11000 14000 16000
24000 26000 44000 46000];
```

```
% shrbin=[0 200 200 400 400 600 600 800 800 1000 1000 3000 3000 7000 8000 12000
13000 17000 23000 27000 43000 47000];
% % shrbin=[-500 500 -400 600 -300 700 -200 800 -100 900 0 1000 100 1100 200 1200
300 1300 400 1400 500 1500 1000 3000 4000 6000 9000 11000 14000 16000 24000
26000 44000 46000];
% for i=1:2:length(shrbin)
%     [nn,xbins]=histc(shry,shrbin([i:(i+1)]));%xbins is the bin location
%     [nn,xbinss]=histc(srinity,shrbin([i:(i+1)]));%xbins is the bin location
%     xxbins=[xxbins;xbins];
%     xxbinss=[xxbinss;xbinss];
%     sb=[sb mean(shrbin([i:(i+1)]))];
% end
% xxbins=logical(xxbins);
% xxbinss=logical(xxbinss);
%
% gbin=zeros(length(sb),2);
% gbinn=zeros(length(sb),2);
% hig=[];
% for j=1:length(sb)
%     gbin(j,2)=mean(grwy(xxbins(j,:)));
%     %     gbinn(j,2)=std(grwy(xxbins(j,:)));%./sum((xxbins(j,:))>.5);
%     gbinn(j,2)=std(grwy(xxbins(j,:)))./sum((xxbins(j,:))>.5);
%     gbin(j,1)=mean((60.*.8./8.8).*(phs1tim)./inity(xxbinss(j,:)));
%     %
gbinn(j,1)=std((60.*.8./8.8).*(phs1tim)./inity(xxbinss(j,:)));%./sum((xxbinss(j,:))>.5);
%     gbinn(j,1)=std((60.*.8./8.8).*(phs1tim)./inity(xxbinss(j,:)))./sum((xxbinss(j,:))>.5);
%     [h,significance,ci] =
ttest2(grwy(xxbins(j,:)),(60.*.8./8.8).*(phs1tim)./inity(xxbinss(j,:)),.05)
%     hig=[hig h];
% end
%
% groupnames = {'0-200';'200-400';'400-600';'600-800';'800-1,000';'1,000-3,000';'3,000-
7,000'; '8,000-12,000'; '13,000-17,000'; '23,000-27,000'; '43,000-47,000'};
% figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
% barweb(gbin,gbinn, [], groupnames, [], [], 'Platelets/micron^2-min', bone, 'y',{'I'; 'II'},
1, 'axis')
% axes1=gca
%
%
% set(gca,'FontSize',20);
% set(gca,'position',[0.1893    0.2000    0.69    0.5]);
% annotation(figure11,'textbox','Position',[0.45 -0.05 0.25
0.16],'LineStyle','none','FitHeightToText','off','String',{'Phase'},'Fontsize',35);
% annotation(figure11,'textbox','Position',[0.4416 0.8981 0.2917
0.05093],'LineStyle','none','FitHeightToText','off','String',{'Shear Rate
(1/s)'},'Fontsize',35);
```

```
% % annotation(figure11,'textbox','Position',[0.45 -0.05 0.25
0.16],'LineStyle','none','FitHeightToText','off','String',{'Phase'},'Fontsize',35);
% ticks=get(gca,'xtick');
% set(gca,'xticklabel',[]);
% text(ticks,3.*ones(length(ticks),1),cellstr(groupnames),'rotation',90,'FontSize',20)
% % legend(handles.bars,['d','a','g','k'])
% % text(ticks,[0 0 1.545 2.316 2.558 2.59 2.236
% % 1.782],cellstr({'';'';'*';'*';'*';'*';'*';'*'}),'rotation',90,'FontSize',30)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% shrbin=[4000 8000];
% [nn,xbins]=histc(shry,shrbin);%xbins is the bin location
% % [nn,sxbins]=histc(shry,shrbin);%xbins is the bin location
% lo=logspace(-1,1,20);
% % [nn,xbins]=histc(grwy(xbins==1),lo);%xbins is the bin location
% figure12=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
% [nn,xout]=hist(grwy(xbins==1),50);
% ax1=bar(xout,nn)
% % ax1=bar(lo,nn)
%
% h = findobj(gca,'Type','patch');
% set(h,'FaceColor',[0 0 0],'EdgeColor','w')
%
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% set(gca, 'XScale', 'log')
% grid on
% xlim([.01 10])
% xlabel('Platelets/microns^2-min','FontSize',35);
% ylabel('# Instances Thrombus Growth Rate (black)','FontSize',35);
% %
% xbinsph2=xbins;
% h1 = gca;
% h2 = axes('Position',get(h1,'Position'));
% [nn,xbins]=histc(srinity,shrbin);%xbins is the bin location
% [nn,xout]=hist(grwinit(xbins==1),10);
%
%  ttest2(grwy(xbinsph2==1),grwinit(xbins==1),0.00001,'both','unequal')
%  [h,significance,ci]=
ttest2(grwy(xbinsph2==1),grwinit(xbins==1),0.00001,'both','unequal')
%
% ax2=bar(xout,nn)
% h = findobj(gca,'Type','patch');
% set(h,'FaceColor',[.3 .3 .3],'EdgeColor','w')
```

```
%
set(h2,'YAxisLocation','right','Color','none');%{'0','2,500','5,000','7,500','10,000','12,500','
15,000'}
% ylabel('# Instances Phase I (gray)','FontSize',35);
% set(h2,'XLim',get(h1,'XLim'),'Layer','top','FontSize',30)
% set(gca, 'XScale', 'log')
%
% %% Create doublearrow
% annotation('doublearrow',[0.3877 0.5613],[0.7074
0.7093],'LineWidth',6,'Head1Width',30,'Head1Length',30,'Head2Width',30,'Head2Length
',30);
% %% Create textbox
% annotation('textbox','Position',[0.3275 0.7257 0.09345
0.06351],'LineStyle','none','FontSize',30,'String',{'Phase I'},'FitHeightToText','on');
%   %% Create doublearrow
% annotation('doublearrow',[0.5622 0.8576],[0.7716
0.7726],'LineWidth',6,'Head1Width',30,'Head1Length',30,'Head2Width',30,'Head2Length
',30);
% %% Create textbox
% annotation('textbox','Position',[0.7856 0.8099 0.09345
0.06351],'LineStyle','none','FontSize',30,'String',{'Thrombus Growth
Rate'},'FitHeightToText','on');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
shrbin=[5000 10000];
[nn,xbins]=histc(shry,shrbin);%xbins is the bin location
% [nn,sxbins]=histc(shry,shrbin);%xbins is the bin location
lo=logspace(-1,1,20);
% [nn,xbins]=histc(grwy(xbins==1),lo);%xbins is the bin location
figure12=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
[nn,xout]=hist(10*grwy(xbins==1),50);
ax1=bar(xout,nn)
% ax1=bar(lo,nn)


h = findobj(gca,'Type','patch');
set(h,'FaceColor',[0 0 0],'EdgeColor','w')

set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% set(gca, 'XScale', 'log')
grid on
xlim([-5 45])
xlabel('Growth Rate (\mum^3/\mum^2-min)','FontSize',35);
ylabel('# Instances','FontSize',35);
```

```
hold on
%r^2=.98;
ftype = fittype('a1.*exp(-((x-b1)./c1).^2)');
[fd gof]=fit(xout',nn',ftype,'StartPoint',[348.3 1.9 1.268],'Lower',[50 1 1],'Robust','on');
gof
dat_std=std(10*grwy(xbins==1));
% a1 =      348.3  ;
% b1 =       1.91  ;
% c1 =      1.268  ;
a1 =      fd.a1 ;
b1 =       fd.b1 ;
c1 =      fd.c1 ;
n_gauss=a1.*exp(-((xout-b1)./c1).^2);
plot(xout,n_gauss,'k--','linewidth',5)
plot([b1+std(10*grwy(xbins==1)) b1+std(10*grwy(xbins==1))],[0 max(nn)+100],'k--
','linewidth',5)
plot([b1-std(10*grwy(xbins==1)) b1-std(10*grwy(xbins==1))],[0 max(nn)+100],'k--
','linewidth',5)
ylim([0 500])
%
%
% h1 = gca;
% h2 = axes('Position',get(h1,'Position'));
% [nn,xbins]=histc(srinity,shrbin);%xbins is the bin location
% [nn,xout]=hist(grwinit,10);
%
%
%
% ax2=bar(xout,nn)
% h = findobj(gca,'Type','patch');
% set(h,'FaceColor',[.3 .3 .3],'EdgeColor','w')
%
set(h2,'YAxisLocation','right','Color','none');%{'0','2,500','5,000','7,500','10,000','12,500','
15,000'}
% ylabel('# Instances Phase I (gray)','FontSize',35);
% set(h2,'XLim',get(h1,'XLim'),'Layer','top','FontSize',30)
% % set(gca, 'XScale', 'log')
%
% %% Create doublearrow
% annotation('doublearrow',[0.3877 0.5613],[0.7074
0.7093],'LineWidth',6,'Head1Width',30,'Head1Length',30,'Head2Width',30,'Head2Length
',30);
% %% Create textbox
% annotation('textbox','Position',[0.3275 0.7257 0.09345
0.06351],'LineStyle','none','FontSize',30,'String',{'Phase I'},'FitHeightToText','on');
%   %% Create doublearrow
```

```
% annotation('doublearrow',[0.5622 0.8576],[0.7716
0.7726],'LineWidth',6,'Head1Width',30,'Head1Length',30,'Head2Width',30,'Head2Length
',30);
% %% Create textbox
% annotation('textbox','Position',[0.7856 0.8099 0.09345
0.06351],'LineStyle','none','FontSize',30,'String',{'Thrombus Growth
Rate'},'FitHeightToText','on');

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% shrbin=[10 300];
% [nn,xbins]=histc(shry,shrbin);%xbins is the bin location
% % [nn,sxbins]=histc(shry,shrbin);%xbins is the bin location
% lo=logspace(-1,1,20);
% % [nn,xbins]=histc(grwy(xbins==1),lo);%xbins is the bin location
% figure12=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
% [nn,xout]=hist(grwy(xbins==1),50);
% ax1=bar(xout,nn)
% % ax1=bar(lo,nn)
% xbinsph2=xbins;
% h = findobj(gca,'Type','patch');
% set(h,'FaceColor',[0 0 0],'EdgeColor','w')
%
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2000    0.7157    0.7250]);
% % set(gca, 'XScale', 'log')
% grid on
% xlim([-1 1])
% xlabel('Growth Rate (Platelets/\mum^2-min)','FontSize',35);
% ylabel('# Instances Thrombus Growth Rate (black)','FontSize',35);
% %
%
% h1 = gca;
% h2 = axes('Position',get(h1,'Position'));
% [nn,xbins]=histc(srinity,shrbin);%xbins is the bin location
% [nn,xout]=hist(grwinit(xbins==1),10);
%
%  [h,significance,ci]= ttest2(grwy(xbinsph2==1),grwinit(xbins==1),0.5,'both','unequal')
%
% ax2=bar(xout,nn)
% h = findobj(gca,'Type','patch');
% set(h,'FaceColor',[.3 .3 .3],'EdgeColor','w')
```

```
%
set(h2,'YAxisLocation','right','Color','none');%{'0','2,500','5,000','7,500','10,000','12,500','
15,000'}
% ylabel('# Instances Phase I (gray)','FontSize',35);
% set(h2,'XLim',get(h1,'XLim'),'Layer','top','FontSize',30)
% % set(gca, 'XScale', 'log')
%
% %% Create doublearrow
% annotation('doublearrow',[0.3877 0.5613],[0.7074
0.7093],'LineWidth',6,'Head1Width',30,'Head1Length',30,'Head2Width',30,'Head2Length
',30);
% %% Create textbox
% annotation('textbox','Position',[0.3275 0.7257 0.09345
0.06351],'LineStyle','none','FontSize',30,'String',{'Phase I'},'FitHeightToText','on');
%   %% Create doublearrow
% annotation('doublearrow',[0.5622 0.8576],[0.7716
0.7726],'LineWidth',6,'Head1Width',30,'Head1Length',30,'Head2Width',30,'Head2Length
',30);
% %% Create textbox
% annotation('textbox','Position',[0.7856 0.8099 0.09345
0.06351],'LineStyle','none','FontSize',30,'String',{'Thrombus Growth
Rate'},'FitHeightToText','on');
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% shrbin=logspace(1,5,100)
% sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
% if (sum(shry<-10000)>0)
%     sb=[mean(shry(shry<shrbin(2))) sb];%x bin values
% else
%     sb=[-20000 sb];
% end
% if (sum(shry>shrbin(length(shrbin)-1))>0)
%     sb=[sb mean(shry(shry>shrbin(length(shrbin)-1)))];%x bin values
% else
%     sb=[sb shrbin(length(shrbin)-1)+10000];
% end
%
%
% [nn,xbins]=histc(shry,shrbin);%xbins is the bin location
% gbin=[];
% gbinn=[];
% sb2=[];
```

```
% for i=[1:length(sb)]
%     if (sum(grwy(xbins==i))~=0)
%         gbin=[gbin (mean(grwy(xbins==i)))];%averaged growth
%         gbinn=[gbinn std((grwy(xbins==i)))];%./sum(xbins==i)];%averaged growth
%         sb2=[sb2 sb(i)];
%     end
% end
% figure
%
hh=semilogx(sb2,gbin,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','Marke
rSize',10,'LineStyle','none');
% hold on
%
hh=semilogx(sb_ph1,gbin_ph1,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','
MarkerSize',10,'LineStyle','none');
% plot([100 800 1500],[.0215 .0615 .10075],'Marker','h','MarkerEdgeColor',[.2 .2
.2],'MarkerFaceColor',[.2 .2 .2],'MarkerSize',10,'LineStyle','none');% Aleveduou 93
% plot([212 1690 3380],[0.067794 0.38826 1.5936],'Marker','p','MarkerEdgeColor',[.4 .4
.4],'MarkerFaceColor',[.4 .4 .4],'MarkerSize',15,'LineStyle','none');%Badimon 86
% plot([420 2600 10500 32000],[.25 1.09375 1.78125
.78125],'Marker','<','MarkerEdgeColor',[.2 .2 .2],'MarkerFaceColor',[.2 .2
.2],'MarkerSize',15,'LineStyle','none');%barstad_94
% plot([420 2600 10500 32000],[.1875 .625 1.5 .625],'Marker','>','MarkerEdgeColor',[.2
.2 .2],'MarkerFaceColor',[.2 .2 .2],'MarkerSize',15,'LineStyle','none');%barstad_96
% plot([11500 10000 24200 32700 42100 42200 40900],[1.2525 0.825 1.41 1.8 2.31
1.7775 2.7],'Marker','^','MarkerEdgeColor',[.2 .2 .2],'MarkerFaceColor',[.2 .2
.2],'MarkerSize',15,'LineStyle','none');%Flannery
% plot([500],[.0125],'Marker','v','MarkerEdgeColor',[.4 .4 .4],'MarkerFaceColor',[.4 .4
.4],'MarkerSize',10,'LineStyle','none');%Hubbel 86
% plot([265 630 2100],[0.01998 0.04998 0.36],'Marker','d','MarkerEdgeColor',[.6 .6
.6],'MarkerFaceColor',[.6 .6 .6],'MarkerSize',15,'LineStyle','none');%Markou 93a
% plot([265 630 900 2120 2400 7500],[0.1155 0.298 0.8255 1.103 1.2295
1.6855],'Marker','s','MarkerEdgeColor',[.6 .6 .6],'MarkerFaceColor',[.4 .4
.4],'MarkerSize',15,'LineStyle','none');%Markou 93b
% plot([100 650 2600],[0.036 0.108 0.396],'Marker','x','MarkerEdgeColor',[.3 .3
.3],'MarkerFaceColor',[.3 .3 .3],'MarkerSize',15,'LineStyle','none');%Sakariassen 90
%
%
%
%
%
% plot(sb2,gbinn+gbin,'LineStyle','--','color',[.4 .4 .4]);
% plot(sb2,gbin-gbinn,'LineStyle','--','color',[.4 .4 .4]);
% plot(sb_ph1,gbinn_ph1+gbin_ph1,'LineStyle','-.','color',[.4 .4 .4]);
% plot(sb_ph1,gbin_ph1-gbinn_ph1,'LineStyle','-.','color',[.4 .4 .4]);
% % plot(sb_ph1,gbinn_ph1+gbin_ph1,'LineStyle','--');
```

```
% % plot(sb_ph1,gbin_ph1-gbinn_ph1,'LineStyle','--');
% %
% %
hh=errorbar(sb2,gbin,gbinn,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',10,'LineWidth',3);
% % hold on
% %
plot(exprs,exprd,'Linestyle','none','Marker','x','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',10);
% xlim([10 100000]);
% legend('Current Thrombus Growth Rate','Current Phase I','Aleveduou 93','Badimon 86','Barstad 94','Barstad 96','Flannery 04','Hubbel 86','Markou 93','Markou 93b','Sakariassen 90');
% % errorbar_tick(hh)
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
% grid on
% xlabel('Shear Rate (1/s)','FontSize',35);
% ylabel('Growth Rate (Platelets/\mum^2-min)','FontSize',35);

ku1=sb2((sb2<=6000)&(sb2>0));
ku2=gbin((sb2<=6000)&(sb2>0));

ku3=shry((shry<=6000)&(shry>500));
ku4=grwy((shry<=6000)&(shry>500));

ku5=sb2((sb2>=6000)&(sb2<100000));
ku6=gbin((sb2>=6000)&(sb2<100000));

ku5=sb2((sb2>=0)&(sb2<100000));
ku6=gbin((sb2>=0)&(sb2<100000));
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% shrbin=[8000:1000:100000];
% sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
% if (sum(shry<-10000)>0)
%     sb=[mean(shry(shry<shrbin(2))) sb];%x bin values
% else
%     sb=[-20000 sb];
% end
% if (sum(shry>shrbin(length(shrbin)-1))>0)
%     sb=[sb mean(shry(shry>shrbin(length(shrbin)-1)))];%x bin values
```

```
% else
%     sb=[sb shrbin(length(shrbin)-1)+10000];
% end
%
%
% [nn,xbins]=histc(shry,shrbin);%xbins is the bin location
% gbin=[];
% gbinn=[];
% sb2=[];
% for i=[1:length(sb)]
%     if (sum(grwy(xbins==i))~=0)
%         gbin=[gbin (mean(grwy(xbins==i)))];%averaged growth
%         gbinn=[gbinn std((grwy(xbins==i)))];%./sum(xbins==i)];%averaged growth
%         sb2=[sb2 sb(i)];
%     end
% end
% figure
%
hh=semilogx(sb2,gbin,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','Marke
rSize',10,'LineStyle','none');
% hold on
%
hh=semilogx(sb_ph1,gbin_ph1,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','
MarkerSize',10,'LineStyle','none');
% plot([100 800 1500],[.0215 .0615 .10075],'Marker','h','MarkerEdgeColor',[.2 .2
.2],'MarkerFaceColor',[.2 .2 .2],'MarkerSize',10,'LineStyle','none');% Aleveduou 93
% plot([212 1690 3380],[0.067794 0.38826 1.5936],'Marker','p','MarkerEdgeColor',[.4 .4
.4],'MarkerFaceColor',[.4 .4 .4],'MarkerSize',15,'LineStyle','none');%Badimon 86
% plot([420 2600 10500 32000],[.25 1.09375 1.78125
.78125],'Marker','<','MarkerEdgeColor',[.2 .2 .2],'MarkerFaceColor',[.2 .2
.2],'MarkerSize',15,'LineStyle','none');%barstad_94
% plot([420 2600 10500 32000],[.1875 .625 1.5 .625],'Marker','>','MarkerEdgeColor',[.2
.2 .2],'MarkerFaceColor',[.2 .2 .2],'MarkerSize',15,'LineStyle','none');%barstad_96
% plot([11500 10000 24200 32700 42100 42200 40900],[1.2525 0.825 1.41 1.8 2.31
1.7775 2.7],'Marker','^','MarkerEdgeColor',[.2 .2 .2],'MarkerFaceColor',[.2 .2
.2],'MarkerSize',15,'LineStyle','none');%Flannery
% plot([500],[.0125],'Marker','v','MarkerEdgeColor',[.4 .4 .4],'MarkerFaceColor',[.4 .4
.4],'MarkerSize',10,'LineStyle','none');%Hubbel 86
% plot([265 630 2100],[0.01998 0.04998 0.36],'Marker','d','MarkerEdgeColor',[.6 .6
.6],'MarkerFaceColor',[.6 .6 .6],'MarkerSize',15,'LineStyle','none');%Markou 93a
% plot([265 630 900 2120 2400 7500],[0.1155 0.298 0.8255 1.103 1.2295
1.6855],'Marker','s','MarkerEdgeColor',[.6 .6 .6],'MarkerFaceColor',[.4 .4
.4],'MarkerSize',15,'LineStyle','none');%Markou 93b
% plot([100 650 2600],[0.036 0.108 0.396],'Marker','x','MarkerEdgeColor',[.3 .3
.3],'MarkerFaceColor',[.3 .3 .3],'MarkerSize',15,'LineStyle','none');%Sakariassen 90
%
```

```
%
%
%
%
% plot(sb2,gbinn+gbin,'LineStyle','--','color',[.4 .4 .4]);
% plot(sb2,gbin-gbinn,'LineStyle','--','color',[.4 .4 .4]);
% plot(sb_ph1,gbinn_ph1+gbin_ph1,'LineStyle','-.','color',[.4 .4 .4]);
% plot(sb_ph1,gbin_ph1-gbinn_ph1,'LineStyle','-.','color',[.4 .4 .4]);
% % plot(sb_ph1,gbinn_ph1+gbin_ph1,'LineStyle','--');
% % plot(sb_ph1,gbin_ph1-gbinn_ph1,'LineStyle','--');
% %
% %
hh=errorbar(sb2,gbin,gbinn,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','Mar
kerSize',10,'LineWidth',3);
% % hold on
% %
plot(exprs,exprd,'Linestyle','none','Marker','x','MarkerEdgeColor','k','MarkerFaceColor','k'
,'MarkerSize',10);
% xlim([10 100000]);
% legend('Current Thrombus Growth Rate','Current Phase I','Aleveduou 93','Badimon
86','Barstad 94','Barstad 96','Flannery 04','Hubbel 86','Markou 93','Markou
93b','Sakariassen 90');
% % errorbar_tick(hh)
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
% grid on
% xlabel('Shear Rate (1/s)','FontSize',35);
% ylabel('Growth Rate (Platelets/\mum^2-min)','FontSize',35);
%
% ku1=sb2((sb2<=6000)&(sb2>0));
% ku2=gbin((sb2<=6000)&(sb2>0));
%
% ku3=shry((shry<=6000)&(shry>500));
% ku4=grwy((shry<=6000)&(shry>500));
%
% ku5=sb2((sb2>=8000)&(sb2<100000));
% ku6=gbin((sb2>=8000)&(sb2<100000));
%
% ku5=sb2((sb2>=0)&(sb2<50000));
% ku6=gbin((sb2>=0)&(sb2<50000));
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```

```
% shrbin=logspace(1,5,100)
% sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
% if (sum(shry<-10000)>0)
%     sb=[mean(shry(shry<shrbin(2))) sb];%x bin values
% else
%     sb=[-20000 sb];
% end
% if (sum(shry>shrbin(length(shrbin)-1))>0)
%     sb=[sb mean(shry(shry>shrbin(length(shrbin)-1)))];%x bin values
% else
%     sb=[sb shrbin(length(shrbin)-1)+10000];
% end
%
%
% [nn,xbins]=histc(shry,shrbin);%xbins is the bin location
% gbin=[];
% gbinn=[];
% sb2=[];
% for i=[1:length(sb)]
%     if (sum(grwy(xbins==i))~=0)
%         gbin=[gbin (mean(grwy(xbins==i)))];%averaged growth
%         gbinn=[gbinn std((grwy(xbins==i)))];%./sum(xbins==i)];%averaged growth
%         sb2=[sb2 sb(i)];
%     end
% end
% figure
% %
hh=semilogx(sb2,gbin,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','Marke
rSize',10,'LineStyle','none');
% hold on
%
hh=semilogx(sb_ph1,gbin_ph1,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','
MarkerSize',10,'LineStyle','none');
% plot([100 800 1500],[.0215 .0615 .10075],'Marker','h','MarkerEdgeColor',[.2 .2
.2],'MarkerFaceColor',[.2 .2 .2],'MarkerSize',10,'LineStyle','none');% Aleveduou 93
% plot([212 1690 3380],[0.067794 0.38826 1.5936],'Marker','p','MarkerEdgeColor',[.4 .4
.4],'MarkerFaceColor',[.4 .4 .4],'MarkerSize',15,'LineStyle','none');%Badimon 86
% plot([420 2600 10500 32000],[.25 1.09375 1.78125
.78125],'Marker','<','MarkerEdgeColor',[.2 .2 .2],'MarkerFaceColor',[.2 .2
.2],'MarkerSize',15,'LineStyle','none');%barstad_94
% plot([420 2600 10500 32000],[.1875 .625 1.5 .625],'Marker','>','MarkerEdgeColor',[.2
.2 .2],'MarkerFaceColor',[.2 .2 .2],'MarkerSize',15,'LineStyle','none');%barstad_96
% plot([11500 10000 24200 32700 42100 42200 40900],[1.2525 0.825 1.41 1.8 2.31
1.7775 2.7],'Marker','^','MarkerEdgeColor',[.2 .2 .2],'MarkerFaceColor',[.2 .2
.2],'MarkerSize',15,'LineStyle','none');%Flannery
```

```
% plot([500],[.0125],'Marker','v','MarkerEdgeColor',[.4 .4 .4],'MarkerFaceColor',[.4 .4
.4],'MarkerSize',10,'LineStyle','none');%Hubbel 86
% plot([265 630 2100],[0.01998 0.04998 0.36],'Marker','d','MarkerEdgeColor',[.6 .6
.6],'MarkerFaceColor',[.6 .6 .6],'MarkerSize',15,'LineStyle','none');%Markou 93a
% plot([265 630 900 2120 2400 7500],[0.1155 0.298 0.8255 1.103 1.2295
1.6855],'Marker','s','MarkerEdgeColor',[.6 .6 .6],'MarkerFaceColor',[.4 .4
.4],'MarkerSize',15,'LineStyle','none');%Markou 93b
% plot([100 650 2600],[0.036 0.108 0.396],'Marker','x','MarkerEdgeColor',[.3 .3
.3],'MarkerFaceColor',[.3 .3 .3],'MarkerSize',15,'LineStyle','none');%Sakariassen 90
%
%
%
% plot(sb_ph1,gbinn_ph1+gbin_ph1,'LineStyle','-.','color',[.4 .4 .4]);
% plot(sb_ph1,gbin_ph1-gbinn_ph1,'LineStyle','-.','color',[.4 .4 .4]);
% % plot(sb_ph1,gbinn_ph1+gbin_ph1,'LineStyle','--');
% % plot(sb_ph1,gbin_ph1-gbinn_ph1,'LineStyle','--');
% %
% %
hh=errorbar(sb2,gbin,gbinn,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','Mar
kerSize',10,'LineWidth',3);
% % hold on
% %
plot(exprs,exprd,'Linestyle','none','Marker','x','MarkerEdgeColor','k','MarkerFaceColor','k'
,'MarkerSize',10);
% xlim([10 100000]);
% legend('Current Phase I','Aleveduou 93','Badimon 86','Barstad 94','Barstad
96','Flannery 04','Hubbel 86','Markou 93','Markou 93b','Sakariassen 90');
% % errorbar_tick(hh)
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
% grid on
% xlabel('Shear Rate (1/s)','FontSize',35);
% ylabel('Growth Rate (Platelets/\mum^2-min)','FontSize',35);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

shrbin=logspace(1,5,100)
sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
if (sum(shry<-10000)>0)
   sb=[mean(shry(shry<shrbin(2))) sb];%x bin values
else
   sb=[-20000 sb];
end
if (sum(shry>shrbin(length(shrbin)-1))>0)
   sb=[sb mean(shry(shry>shrbin(length(shrbin)-1)))];%x bin values
```

```
else
   sb=[sb shrbin(length(shrbin)-1)+10000];
end


[nn,xbins]=histc(shry,shrbin);%xbins is the bin location
gbin=[];
gbinn=[];
sb2=[];
for i=[1:length(sb)]
   if (sum(grwy(xbins==i))~=0)
      gbin=[gbin (mean(grwy(xbins==i)))];%averaged growth
      gbinn=[gbinn std((grwy(xbins==i)))];%./sum(xbins==i)];%averaged growth
      sb2=[sb2 sb(i)];
   end
end
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
hh=semilogx(sb2,gbin,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','Marke
rSize',10,'LineStyle','none');
hold on
hh=semilogx(sb2,-gbinn+gbin,'LineStyle','--','color','k');
%
hh=semilogx(sb_ph1,gbin_ph1,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','
MarkerSize',10,'LineStyle','none');
plot([100 800 1500],[.0215 .0615 .10075],'Marker','h','MarkerEdgeColor',[.2 .2
.2],'MarkerFaceColor',[.2 .2 .2],'MarkerSize',10,'LineStyle','none');% Aleveduou 93
plot([212 1690 3380],[0.067794 0.38826 1.5936],'Marker','p','MarkerEdgeColor',[.4 .4
.4],'MarkerFaceColor',[.4 .4 .4],'MarkerSize',15,'LineStyle','none');%Badimon 86
plot([420 2600 10500 32000],[.25 1.09375 1.78125
.78125],'Marker','<','MarkerEdgeColor',[.2 .2 .2],'MarkerFaceColor',[.2 .2
.2],'MarkerSize',15,'LineStyle','none');%barstad_94
plot([420 2600 10500 32000],[.1875 .625 1.5 .625],'Marker','>','MarkerEdgeColor',[.2 .2
.2],'MarkerFaceColor',[.2 .2 .2],'MarkerSize',15,'LineStyle','none');%barstad_96
plot([11500 10000 24200 32700 42100 42200 40900],[1.2525 0.825 1.41 1.8 2.31 1.7775
2.7],'Marker','^','MarkerEdgeColor',[.2 .2 .2],'MarkerFaceColor',[.2 .2
.2],'MarkerSize',15,'LineStyle','none');%Flannery
% plot([500],[.0125],'Marker','v','MarkerEdgeColor',[.4 .4 .4],'MarkerFaceColor',[.4 .4
.4],'MarkerSize',10,'LineStyle','none');%Hubbel 86
plot([265 630 2100],[0.01998 0.04998 0.36],'Marker','d','MarkerEdgeColor',[.6 .6
.6],'MarkerFaceColor',[.6 .6 .6],'MarkerSize',15,'LineStyle','none');%Markou 93a
plot([265 630 900 2120 2400 7500],[0.1155 0.298 0.8255 1.103 1.2295
1.6855],'Marker','s','MarkerEdgeColor',[.6 .6 .6],'MarkerFaceColor',[.4 .4
.4],'MarkerSize',15,'LineStyle','none');%Markou 93b
plot([100 650 2600],[0.036 0.108 0.396],'Marker','x','MarkerEdgeColor',[.3 .3
.3],'MarkerFaceColor',[.3 .3 .3],'MarkerSize',15,'LineStyle','none');%Sakariassen 90
```

```
hh=semilogx(sb2,gbinn+gbin,'LineStyle','--','color','k');




% plot(sb2,gbinn+gbin,'LineStyle','--','color',[.4 .4 .4]);
% plot(sb2,gbin-gbinn,'LineStyle','--','color',[.4 .4 .4]);
% plot(sb_ph1,gbinn_ph1+gbin_ph1,'LineStyle','--');
% plot(sb_ph1,gbin_ph1-gbinn_ph1,'LineStyle','--');
%
%
hh=errorbar(sb2,gbin,gbinn,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','Mar
kerSize',10,'LineWidth',3);
% hold on
%
plot(exprs,exprd,'Linestyle','none','Marker','x','MarkerEdgeColor','k','MarkerFaceColor','k'
,'MarkerSize',10);
xlim([10 100000]);
ylim([-1 3])
legend('Mean','Standard Deviation','Alevriadou 93','Badimon 86','Barstad 94','Barstad
96','Ku 07','Markou 93','Markou 93b','Sakariassen 90');
% errorbar_tick(hh)
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
grid on
xlabel('Shear Rate (1/s)','FontSize',35);
ylabel('Thrombus Growth Rate (Platelets/\mum^2-min)','FontSize',35);

figure
loglog(sb,nn([1:length(sb)]),'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','
MarkerSize',10,'LineStyle','none');
xlim([10 100000]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
grid on
xlabel('Shear Rate (1/s)','FontSize',35);
ylabel('Number of Instances in Bin','FontSize',35);


%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% grwy=grwy(shry>0);
% rimy=rimy(shry>0);
% xxy=xxy(shry>0);
```

```
%
% grwdat=grwdat(shry>0);
% shry=shry(shry>0);
% % shl=grwy.*2.*rimy./(shry.*.2)./60./(.002./(8));
% % xstar=((shry.*2e-13).*((xxy+min(xxy))*10^-6)./(shry.*((2.*rimy*10^(-
6)).^3))).^(1./3);
% %
% % shl=grwy;
% % xstar=((.002./8.8)/1.8575)./((xxy-min(xxy)+1e-11).^(1/3));
% % shl=shl./xstar;
% % figure
% % loglog(shry,shl,'.')
%
% Re=shry.*1025.*((rimy*10^-6).^2)./.0038
%
% % grwy=shl;
% shrbin=logspace(0,3,300)
% % shrbin=logspace(1,5,100)
% % shrbin=logspace(-4,-1,100)
% % shrbin=[1:1:100];
% sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
% if (sum(Re<-10000)>0)
%     sb=[mean(Re(Re<shrbin(2))) sb];%x bin values
% else
%     sb=[-20000 sb];
% end
% if (sum(Re>shrbin(length(shrbin)-1))>0)
%     sb=[sb mean(Re(Re>shrbin(length(shrbin)-1)))];%x bin values
% else
%     sb=[sb shrbin(length(shrbin)-1)+10000];
% end
%
%
% [nn,xbins]=histc(Re,shrbin);%xbins is the bin location
% gbin=[];
% gbinn=[];
% sb2=[];
% for i=[1:length(sb)]
%     if (sum(grwy(xbins==i))~=0)
%         gbin=[gbin ((mean(grwy(xbins==i))))];%averaged growth
%         gbinn=[gbinn std((grwy(xbins==i)))];%./sum(xbins==i)];%averaged growth
%         sb2=[sb2 sb(i)];
%     end
% end
%
% figure
```

```
%
hh=semilogx(sb2,gbin,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','Marke
rSize',10,'LineStyle','none');
% hold on
% plot(sb2,gbinn+gbin,'LineStyle','--','color',[.4 .4 .4]);
% plot(sb2,gbin-gbinn,'LineStyle','--','color',[.4 .4 .4]);
%
% % plot(sb_ph1,gbinn_ph1+gbin_ph1,'LineStyle','--');
% % plot(sb_ph1,gbin_ph1-gbinn_ph1,'LineStyle','--');
% %
% %
hh=errorbar(sb2,gbin,gbinn,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','Mar
kerSize',10,'LineWidth',3);
% % hold on
% %
plot(exprs,exprd,'Linestyle','none','Marker','x','MarkerEdgeColor','k','MarkerFaceColor','k'
,'MarkerSize',10);
% xlim([1 100]);
% ylim([-1 5]);
% legend('Current Thrombus Growth Rate','STD');
% % errorbar_tick(hh)
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
% grid on
% xlabel('Re','FontSize',35);
% ylabel('Growth Rate (Platelets/\mum^2-min)','FontSize',35);
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
% figure
```

```
%
loglog(sb,nn([1:length(sb)]),'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','
MarkerSize',10,'LineStyle','none');
% xlim([1 100]);
% ku1=sb2((sb2<=100)&(sb2>1));
% ku2=gbin((sb2<=100)&(sb2>1));
%
% ku3=shry((shry<=100)&(shry>1));
% ku4=grwy((shry<=100)&(shry>1));

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% grwy=grwy(shry>0);
% rimy=rimy(shry>0);
% xxy=xxy(shry>0);
%
% grwdat=grwdat(shry>0);
% shry=shry(shry>0);
% % shl=grwy.*2.*rimy./(shry.*.2)./60./(.002./(8));
% % xstar=((shry.*2e-13).*((xxy+min(xxy))*10^-6)./(shry.*((2.*rimy*10^(-
6)).^3))).^(1./3);
% %
% % shl=grwy;
% % xstar=((.002./8.8)/1.8575)./((xxy-min(xxy)+1e-11).^(1/3));
% % shl=shl./xstar;
% % figure
% % loglog(shry,shl,'.')
%
% Re=(1e-6).*rimy.*(shry./((2e-11).*(xxy-min(xxy)).*1e-6)).^(1./3);
%
% % grwy=shl;
% % shrbin=logspace(0,3,300)
% % shrbin=logspace(1,5,100)
% % shrbin=logspace(-4,-1,100)
% shrbin=[10:.5:100];
% sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
% if (sum(Re<-10000)>0)
%     sb=[mean(Re(Re<shrbin(2))) sb];%x bin values
% else
%     sb=[-20000 sb];
% end
% if (sum(Re>shrbin(length(shrbin)-1))>0)
%     sb=[sb mean(Re(Re>shrbin(length(shrbin)-1)))];%x bin values
% else
%     sb=[sb shrbin(length(shrbin)-1)+10000];
```

```
% end
%
%
% [nn,xbins]=histc(Re,shrbin);%xbins is the bin location
% gbin=[];
% gbinn=[];
% sb2=[];
% for i=[1:length(sb)]
%     if (sum(grwy(xbins==i))~=0)
%         gbin=[gbin ((mean(grwy(xbins==i))))];%averaged growth
%         gbinn=[gbinn std((grwy(xbins==i)))];%./sum(xbins==i)];%averaged growth
%         sb2=[sb2 sb(i)];
%     end
% end
%
% figure
%
hh=plot(sb2,gbin,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','none','MarkerSize
',10,'LineStyle','none');
% hold on
% plot(sb2,gbinn+gbin,'LineStyle','--','color',[.4 .4 .4]);
% plot(sb2,gbin-gbinn,'LineStyle','--','color',[.4 .4 .4]);
%
% % plot(sb_ph1,gbinn_ph1+gbin_ph1,'LineStyle','--');
% % plot(sb_ph1,gbin_ph1-gbinn_ph1,'LineStyle','--');
% %
% %
hh=errorbar(sb2,gbin,gbinn,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','Mar
kerSize',10,'LineWidth',3);
% % hold on
% %
plot(exprs,exprd,'Linestyle','none','Marker','x','MarkerEdgeColor','k','MarkerFaceColor','k'
,'MarkerSize',10);
% % xlim([1 100]);
% ylim([-1 5]);
% legend('Current Thrombus Growth Rate','STD');
% % errorbar_tick(hh)
% set(gca,'FontSize',30);
% set(gca,'position',[0.1893    0.2500    0.7157    0.68]);
% grid on
% xlabel('nondimensional','FontSize',35);
% ylabel('Growth Rate (Platelets/\mum^2-min)','FontSize',35);
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Appendix J: Occlusion Time Predictions (Chapter 3)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Predicting occlusion time in chapter 3
%David Bark
%11/11/10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
st=[0:.1:100];
rr=(3e-3).*(100-st)/100+1e-6;
% st(length(st))=[];
% r=rr([1:length(st)]);
r=rr;
Q=(2e-6)*(1.-(1.9e-2)*st+(6.2e-4)*(st.^2)-(5.2e-6).*(st.^3));
% Q=(1.736e-6)*pi*((.75e-3)^2)*(1.-(1.9e-2)*st+(6.2e-4)*(st.^2)-(5.2e-6).*(st.^3));
stl=[0 40 50 60 70 75 85 90 93 95 97 98 99];
Ql=[2.00E-06 2.00E-06 2.00E-06 2.00E-06 2.00E-06 1.90E-06 7.05E-07 3.07E-07
1.44E-07 6.86E-08 2.08E-08 7.35E-09 7.90E-10];
stl2=[75 85 90 93 95 97 98 99];
% Ql2=[1.90E-06 7.05E-07 3.07E-07 1.44E-07 6.86E-08 2.08E-08 7.35E-09 7.90E-10];
% Ql2=(2e-10)*(stl2.^3)+(-4.9932e-8)*(stl2.^2)+(4.0351e-6).*(stl2)+(-.00010449)
Ql2=(2.9623e-009)*(stl2.^2)+(-5.95e-007).*(stl2)+(2.9865e-005);

figure
plot(st,Q)
hold on
plot(stl,Ql,'r')
plot(stl2,Ql2,'g')


g=4*(Q)./pi./(r.^3);
g(g>50000)=50000;
figure
plot(st,g)
J=((1e-6)./60).*((.41.*g.^(.45)+0).*double(g<6000)+(-(2.1e-
4).*g+20).*double(g>=6000));
J(J<0)=1e-13;
figure
plot(st,J)
t_lag=718.*g.^(-.2);
figure
plot(st,t_lag)
le=length(rr);
```

```
to=[];
to=[(rr(le-1)-rr(le))./J(le-1)+t_lag(le-1)];
tod=to;
le=le-1;
while le>1
   tod=tod+(rr(le-1)-rr(le))./J(le-1);
   to=[tod to];
   le=le-1;
end
to=to+t_lag([1:(length(t_lag)-1)]);
figure11=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
semilogy(st([1:(length(st)-1)]),to/60,'linestyle','-.','color','k', 'linewidth', 5);
grid on
% plot(st,to/60)
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2500    0.7157    0.68]);

xlabel('% Stenosis by Diameter','FontSize',35);
ylabel('Occlusion Time (minutes)','FontSize',35);
```

Appendix K: C Code for User-Defined Functions in Fluent to Characterize Species Transport Model (Chapter 4)

```
/*Integrate shear over stenosis*/
/*David Bark*/
/*11/11/10*/

#include "udf.h"
#include <math.h>
#include "math.h"
const TABLESIZE=100;/*595*/
real L=2.379e-3;
real ras=.00075;
real curvy[101];/*596*/
real st=.8484;/*8484*/
real vel=.002358;
real stmin;
/*real vel=.045;*/

#define TSTART 0.0 /* field applied at t = tstart */
/*Define porosity from 0-1 in the computational domain based on user defined memory
(C_UDMI(0))*/

DEFINE_DIFFUSIVITY(d_RBC,c,t,i)
{
        real x[ND_ND], r, flux;
        C_CENTROID(x,c,t);
        r=(6./5.)*(.4)*(1-pow(x[1]/.00075,2));
        if (i==0)
        {
                if  (((C_UDMI(c,t,1)>0.25)&&(C_UDMI(c,t,1)<.75)))
                        flux=0.;
                else
                {
                        if (x[0]<-.005)
                                flux= 1025.*(1.6e-4);
                        else
                        {
                                if
((1>C_UDSI(c,t,0))&&(C_STRAIN_RATE_MAG(c,t)<400000))
                                        flux= 1025.*(.15*(4.2e-6)*(4.2e-
6)*(C_UDSI(c,t,0)+C_UDSI(c,t,1))*pow(fabs(1-
(C_UDSI(c,t,0)+C_UDSI(c,t,1))),.8)*C_STRAIN_RATE_MAG(c,t)+1.6e-13);
                                else
```

```
                                        flux= 1.6e-13;
                                }
                        }
                        /*if ((C_UDMI(c,t,1)>2.25)&&(C_UDMI(c,t,1)<3.75))
                                flux=1.e-8;*/
        }
        else
        {
                if (((C_UDMI(c,t,1)>0.25)&&(C_UDMI(c,t,1)<.75)))
                        flux= 0.;
                else
                {
                        if (x[0]<-.005)
                                flux= 1025.*(1.6e-4)*C_UDSI(c,t,0);
                        else
                        {
                                if
((1>C_UDSI(c,t,0))&&(C_STRAIN_RATE_MAG(c,t)<400000))
                                        flux= 1025.*(.15*(4.2e-6)*(4.2e-
6)*C_UDSI(c,t,0)*(C_UDSI(c,t,0)+C_UDSI(c,t,1))*pow(fabs(1-
(C_UDSI(c,t,0)+C_UDSI(c,t,1))),.8)*C_STRAIN_RATE_MAG(c,t)+1.6e-13);
                                else
                                        flux=1.6e-13;
                                /*if ((C_UDMI(c,t,1)>2.25)&&(C_UDMI(c,t,1)<3.75))
                                        flux=1025.*1.e-13;*/
                        }
                }
        }
        if (flux<0)
                flux=1.e-15;
        return flux;
}
DEFINE_DIFFUSIVITY(d_brown,c,t,i)
{
        real x[ND_ND], r, flux;
        C_CENTROID(x,c,t);
        r=(6./5.)*(.4)*(1-pow(x[1]/.00075,2));
        if (i==0)
        {
                if  (((C_UDMI(c,t,1)>0.25)&&(C_UDMI(c,t,1)<.75)))
                        flux=0.;
                else
                {
                        if (x[0]<-.005)
                                flux= 1025.*(1.6e-4);
                        else
```

```
                {
                        if
((1>C_UDSI(c,t,0))&&(C_STRAIN_RATE_MAG(c,t)<400000))
                                flux= 1025.*(.15*(4.2e-6)*(4.2e-
6)*(C_UDSI(c,t,0)+C_UDSI(c,t,1))*pow(fabs(1-
(C_UDSI(c,t,0)+C_UDSI(c,t,1))),.8)*C_STRAIN_RATE_MAG(c,t)+1.6e-13);
                        else
                                flux= 1.6e-13;
                }
        }
        /*if ((C_UDMI(c,t,1)>2.25)&&(C_UDMI(c,t,1)<3.75))
                flux=1.e-8;*/
    }
    else
    {
        if (((C_UDMI(c,t,1)>0.25)&&(C_UDMI(c,t,1)<.75)))
                flux= 0.;
        else
        {
                if (x[0]<-.005)
                        flux= 1025.*(1.6e-4)*C_UDSI(c,t,0);
                else
                {
                        if
((1>C_UDSI(c,t,0))&&(C_STRAIN_RATE_MAG(c,t)<400000))
                                flux= 1025.*(1.6e-13)*C_UDSI(c,t,0);
                        else
                                flux=1.6e-13;
                        /*if ((C_UDMI(c,t,1)>2.25)&&(C_UDMI(c,t,1)<3.75))
                                flux=1025.*1.e-13;*/
                }
        }
    }
    if (flux<0)
        flux=1.e-15;
    return flux;
}
DEFINE_PROFILE(vel_,t,i)
{
    face_t f;
    real x[ND_ND];
    begin_f_loop(f,t)
    {
        F_CENTROID(x,f,t);
        F_PROFILE(f,t,i)=vel*2.*(1.-pow(x[1]/ras,2));/*14.9626*fourier;*/
    }
```

```
        end_f_loop(f,t)
}
DEFINE_PROFILE(vel_ffr,t,i)
{
        face_t f;
        real x[ND_ND], u;
        begin_f_loop(f,t)
        {
                F_CENTROID(x,f,t);
                u=((2.9623e-9)*pow(stmin,2)-(5.95e-7)*stmin+2.9865e-5)*141471.061;
                if (u>.282942121)
                        u=.282942121;
                F_PROFILE(f,t,i)=u*2.*(1-pow(x[1]/ras,2));/*14.9626*fourier;*/
        }
        end_f_loop(f,t)
}
DEFINE_PROFILE(out0,t,i)
{
        face_t f;
        cell_t c;
        Thread *t1;
        begin_f_loop(f,t)
        {
                c=F_C0(f,t);
                t1=THREAD_T0(t);
                F_PROFILE(f,t,i)= C_UDSI(c,t1,i);
        }
        end_f_loop(f,t)
}
DEFINE_SOURCE(k1_simp,c,t,dS,eqn)
{
        real cen, con, conn, source;
        real A[ND_ND],x1[ND_ND],x2[ND_ND],mult[ND_ND];
        real k_rat;
        int n;
        face_t f1;
        cell_t c1;
        Thread *t1,*t2;
        k_rat=1025.*(2.e-4-(1.e-9)*C_STRAIN_RATE_MAG(c,t));/*1025.*1.e-5-(1.e-
9)*1025.*C_STRAIN_RATE_MAG(c,t);*/
        k_rat=1025.*(10000.);
        if (k_rat<(1025.*1.e-5))
                k_rat=1025.*1000000.;
        if((C_UDMI(c,t,1)>2.5))/* &&(C_STRAIN_RATE_MAG(c,t)>1000.)
&&(CURRENT_TIME>(718.*pow(C_STRAIN_RATE_MAG(c,t)+1.e-20,-.2))))     */
        {
```

```
                if (C_UDSI(c,t,1)<0.)
                {
                        source=0.;
                        dS[eqn] =0.;
                }
                else
                {
                        source=-k_rat*C_UDSI(c,t,1);
                        dS[eqn] =-k_rat;
                }
        }
        else
        {
                source = 0.;
                dS[eqn] = 0.;
        }
        /*C_UDMI(c,t,4)=k_rat;*/
        return source;
}
DEFINE_PROFILE(vis_res,t,i)
{
        real x[ND_ND];
        real a;
        cell_t c;
        begin_c_loop(c,t)
        {
                C_CENTROID(x,c,t);
                if(((C_UDMI(c,t,1)<.75)&&(C_UDMI(c,t,1)>.25)))
                        a = 1.0e50;
                else
                {
                        if(((C_UDMI(c,t,1)<3.75)&&(C_UDMI(c,t,1)>2.25)))
                                a = 1.0e50;
                        else
                                a = 0.0;
                /*      if ((C_UDMI(c,t,1)<.49)&&(C_UDMI(c,t,1)>.41))
                                a=(CURRENT_TIME-
C_UDMI(c,t,9))*(1e30)/(C_UDMI(c,t,9)-C_UDMI(c,t,10));
                        else
                                a = 0.0;
                        if (a>1.e30)
                                a=1.e30;*/
                }
                F_PROFILE(c,t,i) = a;
        }
        end_c_loop(c,t)
```

```
}

DEFINE_UDS_FLUX(my_uds_flux,f,t,i)
{
        cell_t c0, c1 = -1;
        Thread *t0, *t1 = NULL;
        real NV_VEC(psi_vec), NV_VEC(A), NV_VEC(dif), flux = 0.0,
x[ND_ND],x1[ND_ND], r, rd;
        real al, del,R;
        c0 = F_C0(f,t);
        t0 = F_C0_THREAD(f,t);
        F_AREA(A, f, t);/* If face lies at domain boundary, use face values; */
        F_CENTROID(x,f,t);

        if (fabs(x[0])<L)
        {
                int currn=((x[0]/L+1.)*TABLESIZE/2.+.5);
                del=.2263*curvy[currn];
                al=7.873/(curvy[currn]+1.e-20);
                r=curvy[currn];
        }
        else
        {
                del=.2263*ras;/*C_UDMI(c0,t0,2);*/
                al=7.873/ras;/*C_UDMI(c0,t0,2);*/
                r=ras;
        }
        /* If face lies IN the domain, use average of adjacent cells. */
        if (BOUNDARY_FACE_THREAD_P(t)) /*Most face values will be available*/
        {
                        flux=F_FLUX(f,t);
        }
        else
        {
                c1 = F_C1(f,t); /* Get cell on other side of face */
                t1 = F_C1_THREAD(f,t);
                if (i==0)
                {

        /*(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))*/
                        if
(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.25)&&(C
_UDMI(c1,t1,1)<.75)))
                        {
                                NV_DS(dif, =, 0.,-1.,0.,*,1025.);
```

```
                flux=NV_DOT(dif, A);
            }
            else
            {
                if
(((C_UDMI(c0,t0,1)>2.25)&&(C_UDMI(c0,t0,1)<3.75))||((C_UDMI(c1,t1,1)>2.25)&&(
C_UDMI(c1,t1,1)<3.75)))
                {
                    NV_DS(dif, =, 0.,0.,0.,*,1025.);
                    flux=NV_DOT(dif, A);
                }
                else
                {

        flux=.5*(C_UDSI_DIFF(c0,t0,i)+C_UDSI_DIFF(c1,t1,i));
                    NV_DS(dif, =, 0.,flux,0.,*,(-al*tanh(-
r*al+al*x[1]+al*del)-al));
                    flux=F_FLUX(f,t)+NV_DOT(dif, A);
                }
            }
        }
        else
        {

        if(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))
            {
                NV_DS(dif, =,0.,-1.,0.,*,1025.);
                flux=NV_DOT(dif, A);
            }
            else
            {

                if
(((C_UDMI(c0,t0,1)>2.25)&&(C_UDMI(c0,t0,1)<3.75))||((C_UDMI(c1,t1,1)>2.25)&&(
C_UDMI(c1,t1,1)<3.75)))
                {
                    NV_DS(dif, =, 0.,0.,0.,*,1025.);
                    if
((C_UDMI(c0,t0,1)>1.25)&&(C_UDMI(c0,t0,1)<2.75))
                    {
                        C_CENTROID(x1,c0,t0);
                        NV_VV(x1,=,x1,-,x);
                        NV_VS(x1,=,x1,/,NV_MAG(x1));
                        flux=-1025.*(1.e-
4)*C_UDSI(c0,t0,1)*NV_DOT(x1,A);
```

```
                                        flux=NV_DOT(dif, A);
                                        C_UDMI(c0,t0,5)=fabs(1025.*(1.e-
4)*C_UDSI(c0,t0,1)*NV_DOT(x1,A));

                                }
                                else
                                {
                                        if
((C_UDMI(c1,t1,1)>1.25)&&(C_UDMI(c1,t1,1)<2.75))
                                        {
                                                C_CENTROID(x1,c1,t1);
                                                NV_VV(x1,=,x1,-,x);
                                        NV_VS(x1,=,x1,/,NV_MAG(x1));
                                        flux=-1025.*(1.e-
4)*C_UDSI(c1,t1,1)*NV_DOT(x1,A);
                                                C_UDMI(c1,t1,5)=fabs(1025.*(1.e-
4)*C_UDSI(c0,t0,1)*NV_DOT(x1,A));
                                                flux=NV_DOT(dif, A);

                                        }
                                        else
                                                flux=NV_DOT(dif, A);
                                }
                        }
                        else
                        {

        flux=.5*(C_UDSI_DIFF(c0,t0,0)+C_UDSI_DIFF(c1,t1,0));
                                        NV_DS(dif, =,
.5*(C_UDSI_G(c0,t0,0)[0]+C_UDSI_G(c1,t1,0)[0]),.5*(C_UDSI_G(c0,t0,0)[1]+C_UDS
I_G(c1,t1,0)[1]),0.,*,(-1.)*flux);
                                        flux=F_FLUX(f,t)+NV_DOT(dif, A);
                        }
                }
        }
        return flux;
}
DEFINE_UDS_FLUX(flux_work,f,t,i)
{
        cell_t c0, c1 = -1;
        Thread *t0, *t1 = NULL;
        real NV_VEC(psi_vec), NV_VEC(A), NV_VEC(dif), flux = 0.0,
x[ND_ND],x1[ND_ND],x2[ND_ND], r, rd;
        real al, del,R, rat;
        c0 = F_C0(f,t);
```

```
t0 = F_C0_THREAD(f,t);
F_AREA(A, f, t);/* If face lies at domain boundary, use face values; */
F_CENTROID(x,f,t);
rat=1.e-4;

if (fabs(x[0])<L)
{
        int currn=((x[0]/L+1.)*TABLESIZE/2.+.5);
        del=.2263*curvy[currn];
        al=7.873/(curvy[currn]+1.e-20);
        r=curvy[currn];
}
else
{
        del=.2263*ras;/*C_UDMI(c0,t0,2);*/
        al=7.873/ras;/*C_UDMI(c0,t0,2);*/
        r=ras;
}
/* If face lies IN the domain, use average of adjacent cells. */
if (BOUNDARY_FACE_THREAD_P(t)) /*Most face values will be available*/
{
                flux=F_FLUX(f,t);
}
else
{
        c1 = F_C1(f,t); /* Get cell on other side of face */
        t1 = F_C1_THREAD(f,t);
        if (i==0)
        {

/*(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))*/
                if
(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.25)&&(C
_UDMI(c1,t1,1)<.75)))
                {
                        NV_DS(dif, =, 0.,-1.,0.,*,1025.);
                        flux=NV_DOT(dif, A);
                }
                else
                {
                        if
(((C_UDMI(c0,t0,1)>2.25)&&(C_UDMI(c0,t0,1)<3.75))||((C_UDMI(c1,t1,1)>2.25)&&(
C_UDMI(c1,t1,1)<3.75)))
                        {
                                NV_DS(dif, =, 0.,0.,0.,*,1025.);
```

```
                                        flux=NV_DOT(dif, A);
                                }
                                else
                                {

        flux=.5*(C_UDSI_DIFF(c0,t0,i)+C_UDSI_DIFF(c1,t1,i));
                                        NV_DS(dif, =, 0.,flux,0.,*,(-al*tanh(-
r*al+al*x[1]+al*del)-al));
                                        flux=F_FLUX(f,t)+NV_DOT(dif, A);
                                }
                        }
                }
                else
                {

        if(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))
                        {
                                NV_DS(dif, =,0.,-1.,0.,*,1025.);
                                flux=NV_DOT(dif, A);
                        }
                        else
                        {
                                if
(((C_UDMI(c0,t0,1)>2.25)&&(C_UDMI(c0,t0,1)<3.75))||((C_UDMI(c1,t1,1)>2.25)&&(
C_UDMI(c1,t1,1)<3.75)))
                                {
                                        NV_DS(dif, =, 0.,0.,0.,*,1025.);
                                        flux=NV_DOT(dif, A);
                                        if
((C_UDMI(c0,t0,1)>1.25)&&(C_UDMI(c0,t0,1)<2.75))
                                        {
                                                C_CENTROID(x1,c0,t0);
                                                NV_VV(x2,=,x1,-,x);
                                                NV_VS(x1,=,x2,/,NV_MAG(x2));
                                                if (NV_DOT(A,x1)>0)
                                                        flux=-
1025.*(rat)*(C_UDSI(c0,t0,1)-NV_DOT(C_UDSI_G(c0,t0,1),x2))*NV_MAG(A);
                                                else
                                                        flux=1025.*(rat)*(C_UDSI(c0,t0,1)-
NV_DOT(C_UDSI_G(c0,t0,1),x2))*NV_MAG(A);
                                                C_UDMI(c0,t0,5)=fabs(1025.*(1.e-
4)*C_UDSI(c0,t0,1));
                                                C_UDMI(c0,t0,4)=1025.*(1.e-4);

                                        }
```

```
                                        if
((C_UDMI(c1,t1,1)>1.25)&&(C_UDMI(c1,t1,1)<2.75))
                                        {
                                                C_CENTROID(x1,c1,t1);
                                                NV_VV(x2,=,x1,-,x);
                                                NV_VS(x1,=,x2,/,NV_MAG(x2));
                                                if (NV_DOT(A,x1)>0)
                                                        flux=-
1025.*(rat)*(C_UDSI(c1,t1,1)-NV_DOT(C_UDSI_G(c1,t1,1),x2))*NV_MAG(A);
                                                else
                                                        flux=1025.*(rat)*(C_UDSI(c1,t1,1)-
NV_DOT(C_UDSI_G(c1,t1,1),x2))*NV_MAG(A);
                                                C_UDMI(c1,t1,5)=fabs(1025.*(1.e-
4)*C_UDSI(c1,t1,1));
                                                C_UDMI(c1,t1,4)=1025.*(1.e-4);
                                        }
                                }
                                else
                                {

        flux=.5*(C_UDSI_DIFF(c0,t0,0)+C_UDSI_DIFF(c1,t1,0));
                                        NV_DS(dif, =,
.5*(C_UDSI_G(c0,t0,0)[0]+C_UDSI_G(c1,t1,0)[0]),.5*(C_UDSI_G(c0,t0,0)[1]+C_UDS
I_G(c1,t1,0)[1]),0.,*,(-1.)*flux);
                                        flux=F_FLUX(f,t)+NV_DOT(dif, A);
                                }
                        }
                }
        }
        return flux;
}
DEFINE_UDS_FLUX(flux_work2,f,t,i)
{
        cell_t c0, c1 = -1;
        Thread *t0, *t1 = NULL;
        real NV_VEC(psi_vec), NV_VEC(A), NV_VEC(dif), flux = 0.0,
x[ND_ND],x1[ND_ND],x2[ND_ND], r, rd;
        real al, del,R, rat;
        c0 = F_C0(f,t);
        t0 = F_C0_THREAD(f,t);
        F_AREA(A, f, t);/* If face lies at domain boundary, use face values; */
        F_CENTROID(x,f,t);
        rat=1.e-4;

        if (fabs(x[0])<L)
        {
```

```
            int currn=((x[0]/L+1.)*TABLESIZE/2.+.5);
            del=.2263*curvy[currn];
            al=7.873/(curvy[currn]+1.e-20);
            r=curvy[currn];
        }
        else
        {
            del=.2263*ras;/*C_UDMI(c0,t0,2);*/
            al=7.873/ras;/*C_UDMI(c0,t0,2);*/
            r=ras;
        }
        /* If face lies IN the domain, use average of adjacent cells. */
        if (BOUNDARY_FACE_THREAD_P(t)) /*Most face values will be available*/
        {
                flux=F_FLUX(f,t);
        }
        else
        {
            c1 = F_C1(f,t); /* Get cell on other side of face */
            t1 = F_C1_THREAD(f,t);
            if (i==0)
            {

        /*(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))*/
                    if
(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.25)&&(C
_UDMI(c1,t1,1)<.75)))
                    {
                        NV_DS(dif, =, 0.,-1.,0.,*,1025.);
                        flux=NV_DOT(dif, A);
                    }
                    else
                    {
                        if
(((C_UDMI(c0,t0,1)>2.25)&&(C_UDMI(c0,t0,1)<3.75))||((C_UDMI(c1,t1,1)>2.25)&&(
C_UDMI(c1,t1,1)<3.75)))
                        {
                            NV_DS(dif, =, 0.,0.,0.,*,1025.);
                            flux=NV_DOT(dif, A);
                        }
                        else
                        {

        flux=.5*(C_UDSI_DIFF(c0,t0,i)+C_UDSI_DIFF(c1,t1,i));
```

```
                                        NV_DS(dif, =, 0.,flux,0.,*,(-al*tanh(-
r*al+al*x[1]+al*del)-al));
                                        flux=F_FLUX(f,t)+NV_DOT(dif, A);
                                }
                        }
                }
                else
                {

        if(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))
                        {
                                NV_DS(dif, =,0.,-1.,0.,*,1025.);
                                flux=NV_DOT(dif, A);
                        }
                        else
                        {
                                if
(((C_UDMI(c0,t0,1)>2.25)&&(C_UDMI(c0,t0,1)<3.75))||((C_UDMI(c1,t1,1)>2.25)&&(
C_UDMI(c1,t1,1)<3.75)))
                                {
                                        NV_DS(dif, =, 0.,0.,0.,*,1025.);
                                        flux=NV_DOT(dif, A);
                                        if
((C_UDMI(c0,t0,1)>1.25)&&(C_UDMI(c0,t0,1)<2.75))
                                        {
                                                flux=1025.*(rat)*C_UDSI(c0,t0,1);
                                                C_UDMI(c0,t0,5)=flux;
                                                NV_DS(dif, =, 0.,flux,0.,*,1.);
                                                flux=F_FLUX(f,t)+NV_DOT(dif, A);
                                                C_UDMI(c0,t0,4)=1025.*(rat);

                                        }
                                        if
((C_UDMI(c1,t1,1)>1.25)&&(C_UDMI(c1,t1,1)<2.75))
                                        {
                                                flux=1025.*(rat)*C_UDSI(c1,t1,1);
                                                C_UDMI(c1,t1,5)=flux;
                                                NV_DS(dif, =, 0.,flux,0.,*,1.);
                                                flux=F_FLUX(f,t)+NV_DOT(dif, A);
                                                C_UDMI(c1,t1,4)=1025.*(rat);;
                                        }
                                }
                                else
                                {
```

```
        flux=.5*(C_UDSI_DIFF(c0,t0,0)+C_UDSI_DIFF(c1,t1,0));
                                        NV_DS(dif, =,
.5*(C_UDSI_G(c0,t0,0)[0]+C_UDSI_G(c1,t1,0)[0]),.5*(C_UDSI_G(c0,t0,0)[1]+C_UDS
I_G(c1,t1,0)[1]),0.,*,(-1.)*flux);
                                        flux=F_FLUX(f,t)+NV_DOT(dif, A);
                                }
                        }
                }
        }
        return flux;
}
DEFINE_UDS_FLUX(flux_work3,f,t,i)
{
        cell_t c0, c1 = -1;
        Thread *t0, *t1 = NULL;
        real NV_VEC(psi_vec), NV_VEC(A), NV_VEC(dif), flux = 0.0,
x[ND_ND],x1[ND_ND],x2[ND_ND], r, rd;
        real al, del,R, rat;
        c0 = F_C0(f,t);
        t0 = F_C0_THREAD(f,t);
        F_AREA(A, f, t);/* If face lies at domain boundary, use face values; */
        F_CENTROID(x,f,t);
        rat=5.e-5;

        if (fabs(x[0])<L)
        {
                int curnn=((x[0]/L+1.)*TABLESIZE/2.+.5);
                del=.2263*curvy[curnn];
                al=7.873/(curvy[curnn]+1.e-20);
                r=curvy[curnn];
        }
        else
        {
                del=.2263*ras;/*C_UDMI(c0,t0,2);*/
                al=7.873/ras;/*C_UDMI(c0,t0,2);*/
                r=ras;
        }
        /* If face lies IN the domain, use average of adjacent cells. */
        if (BOUNDARY_FACE_THREAD_P(t)) /*Most face values will be available*/
        {
                        flux=F_FLUX(f,t);
        }
        else
        {
                c1 = F_C1(f,t); /* Get cell on other side of face */
```

```
                t1 = F_C1_THREAD(f,t);
                if (i==0)
                {

        /*(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))*/
                        if
(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.25)&&(C
_UDMI(c1,t1,1)<.75)))
                        {
                                NV_DS(dif, =, 0.,-1.,0.,*,1025.);
                                flux=NV_DOT(dif, A);
                        }
                        else
                        {
                                if
(((C_UDMI(c0,t0,1)>2.25)&&(C_UDMI(c0,t0,1)<3.75))||((C_UDMI(c1,t1,1)>2.25)&&(
C_UDMI(c1,t1,1)<3.75)))
                                {
                                        NV_DS(dif, =, 0.,0.,0.,*,1025.);
                                        flux=NV_DOT(dif, A);
                                }
                                else
                                {

        flux=.5*(C_UDSI_DIFF(c0,t0,i)+C_UDSI_DIFF(c1,t1,i));
                                        NV_DS(dif, =, 0.,flux,0.,*,(-al*tanh(-
r*al+al*x[1]+al*del)-al));
                                        flux=F_FLUX(f,t)+NV_DOT(dif, A);
                                }
                        }
                }
                else
                {

        if(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))
                        {
                                NV_DS(dif, =,0.,-1.,0.,*,1025.);
                                flux=NV_DOT(dif, A);
                        }
                        else
                        {
                                if
(((C_UDMI(c0,t0,1)>2.25)&&(C_UDMI(c0,t0,1)<3.75))||((C_UDMI(c1,t1,1)>2.25)&&(
C_UDMI(c1,t1,1)<3.75)))
```

```
                {
                        NV_DS(dif, =, 0.,0.,0.,*,1025.);
                        flux=NV_DOT(dif, A);
                        if
((C_UDMI(c0,t0,1)>1.25)&&(C_UDMI(c0,t0,1)<2.75))
                        {
                                C_CENTROID(x1,c0,t0);
                                NV_VV(x2,=,x,-,x1);
                                NV_VS(x1,=,x2,/,NV_MAG(x2));

        flux=1025.*(rat)*(C_UDSI(c0,t0,1)+NV_DOT(C_UDSI_G(c0,t0,1),x2));
                                if (flux<0.)
                                        flux=0.;
                                NV_DS(dif, =, 0.,flux,0.,*,1.);


                                C_UDMI(c0,t0,4)=1025.*(rat);
                                flux=F_FLUX(f,t)+NV_DOT(dif, A);
                                C_UDMI(c0,t0,5)=fabs(NV_DOT(dif, A));
                        }
                        if
((C_UDMI(c1,t1,1)>1.25)&&(C_UDMI(c1,t1,1)<2.75))
                        {
                                C_CENTROID(x1,c1,t1);
                                NV_VV(x2,=,x,-,x1);
                                NV_VS(x1,=,x2,/,NV_MAG(x2));

        flux=1025.*(rat)*(C_UDSI(c1,t1,1)+NV_DOT(C_UDSI_G(c1,t1,1),x2));
                                if (flux<0.)
                                        flux=0.;
                                NV_DS(dif, =, 0.,flux,0.,*,1.);

                                C_UDMI(c1,t1,4)=1025.*(rat);
                                flux=F_FLUX(f,t)+NV_DOT(dif, A);
                                C_UDMI(c1,t1,5)=fabs(NV_DOT(dif, A));
                        }
                }
                else
                {

        flux=.5*(C_UDSI_DIFF(c0,t0,0)+C_UDSI_DIFF(c1,t1,0));
                        NV_DS(dif, =,
.5*(C_UDSI_G(c0,t0,0)[0]+C_UDSI_G(c1,t1,0)[0]),.5*(C_UDSI_G(c0,t0,0)[1]+C_UDS
I_G(c1,t1,0)[1]),0.,*,(-1.)*flux);
                        flux=F_FLUX(f,t)+NV_DOT(dif, A);
                }
```

```
                        }
                }
        }
        return flux;
}
DEFINE_UDS_FLUX(flux_work32,f,t,i)
{
        cell_t c0, c1 = -1;
        Thread *t0, *t1 = NULL;
        real NV_VEC(psi_vec), NV_VEC(A), NV_VEC(dif), flux = 0.0,
x[ND_ND],x1[ND_ND],x2[ND_ND], r, rd;
        real al, del,R, rat;
        c0 = F_C0(f,t);
        t0 = F_C0_THREAD(f,t);
        F_AREA(A, f, t);/* If face lies at domain boundary, use face values; */
        F_CENTROID(x,f,t);
        rat=1.e-4;

        if (fabs(x[0])<L)
        {
                int currn=((x[0]/L+1.)*TABLESIZE/2.+.5);
                del=.2263*curvy[currn];
                al=7.873/(curvy[currn]+1.e-20);
                r=curvy[currn];
        }
        else
        {
                del=.2263*ras;/*C_UDMI(c0,t0,2);*/
                al=7.873/ras;/*C_UDMI(c0,t0,2);*/
                r=ras;
        }
        /* If face lies IN the domain, use average of adjacent cells. */
        if (BOUNDARY_FACE_THREAD_P(t)) /*Most face values will be available*/
        {
                        flux=F_FLUX(f,t);
        }
        else
        {
                c1 = F_C1(f,t); /* Get cell on other side of face */
                t1 = F_C1_THREAD(f,t);
                if (i==0)
                {

        /*(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))*/
```

```
                        if
(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.25)&&(C
_UDMI(c1,t1,1)<.75)))
                        {
                                NV_DS(dif, =, 0.,-1.,0.,*,1025.);
                                flux=NV_DOT(dif, A);
                        }
                        else
                        {
                                if
(((C_UDMI(c0,t0,1)>2.25)&&(C_UDMI(c0,t0,1)<3.75))||((C_UDMI(c1,t1,1)>2.25)&&(
C_UDMI(c1,t1,1)<3.75)))
                                {
                                        NV_DS(dif, =, 0.,0.,0.,*,1025.);
                                        flux=NV_DOT(dif, A);
                                }
                                else
                                {

        flux=.5*(C_UDSI_DIFF(c0,t0,i)+C_UDSI_DIFF(c1,t1,i));
                                        NV_DS(dif, =, 0.,flux,0.,*,(-al*tanh(-
r*al+al*x[1]+al*del)-al));
                                        flux=F_FLUX(f,t)+NV_DOT(dif, A);
                                }
                        }
                }
                else
                {

        if(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))
                        {
                                NV_DS(dif, =,0.,-1.,0.,*,1025.);
                                flux=NV_DOT(dif, A);
                        }
                        else
                        {
                                if
(((C_UDMI(c0,t0,1)>2.25)&&(C_UDMI(c0,t0,1)<3.75))||((C_UDMI(c1,t1,1)>2.25)&&(
C_UDMI(c1,t1,1)<3.75)))
                                {
                                        NV_DS(dif, =, 0.,0.,0.,*,1025.);
                                        flux=NV_DOT(dif, A);
                                        if
((C_UDMI(c0,t0,1)>1.25)&&(C_UDMI(c0,t0,1)<2.75))
                                        {
```

```
                                        C_CENTROID(x1,c0,t0);
                                        if
(CURRENT_TIME>(280*pow(C_STRAIN_RATE_MAG(c0,t0),-.2)))
                                        {

                                        NV_VV(x2,=,x,-,x1);
                                        NV_VS(x1,=,x2,/,NV_MAG(x2));

      /*flux=1025.*(rat)*(C_UDSI(c0,t0,1)+NV_DOT(C_UDSI_G(c0,t0,1),x2));*/
                                        if (NV_DOT(A,x1)>0)

      flux=1025.*(rat)*(C_UDSI(c0,t0,1)+NV_DOT(C_UDSI_G(c0,t0,1),x2))*NV_M
AG(A);
                                        else
                                                flux=-
1025.*(rat)*(C_UDSI(c0,t0,1)+NV_DOT(C_UDSI_G(c0,t0,1),x2))*NV_MAG(A);

                                        C_UDMI(c0,t0,4)=1025.*(rat);
                                        flux=F_FLUX(f,t)+flux;
                                        }
                                }
                                if
((C_UDMI(c1,t1,1)>1.25)&&(C_UDMI(c1,t1,1)<2.75))
                                {
                                        if
(CURRENT_TIME>(280*pow(C_STRAIN_RATE_MAG(c1,t1),-.2)))
                                        {
                                        C_CENTROID(x1,c1,t1);
                                        NV_VV(x2,=,x,-,x1);
                                        NV_VS(x1,=,x2,/,NV_MAG(x2));

      /*flux=1025.*(rat)*(C_UDSI(c1,t1,1)+NV_DOT(C_UDSI_G(c1,t1,1),x2));*/
                                        if (NV_DOT(A,x1)>0)

      flux=1025.*(rat)*(C_UDSI(c1,t1,1)+NV_DOT(C_UDSI_G(c1,t1,1),x2))*NV_M
AG(A);
                                        else
                                                flux=-
1025.*(rat)*(C_UDSI(c1,t1,1)+NV_DOT(C_UDSI_G(c1,t1,1),x2))*NV_MAG(A);

                                        C_UDMI(c1,t1,4)=1025.*(rat);
                                        flux=F_FLUX(f,t)+flux;
                                        }
                                }
                        }
                        else
```

```
                                {

        flux=.5*(C_UDSI_DIFF(c0,t0,0)+C_UDSI_DIFF(c1,t1,0));
                                        NV_DS(dif, =,
.5*(C_UDSI_G(c0,t0,0)[0]+C_UDSI_G(c1,t1,0)[0]),.5*(C_UDSI_G(c0,t0,0)[1]+C_UDS
I_G(c1,t1,0)[1]),0.,*,(-1.)*flux);
                                        flux=F_FLUX(f,t)+NV_DOT(dif, A);
                                }
                        }
                }
        }
        return flux;
}
DEFINE_UDS_FLUX(fluxdif_work32,f,t,i)
{
        cell_t c0, c1 = -1;
        Thread *t0, *t1 = NULL;
        real NV_VEC(psi_vec), NV_VEC(A), NV_VEC(dif), flux = 0.0,
x[ND_ND],x1[ND_ND],x2[ND_ND], r, rd;
        real al, del,R, rat;
        c0 = F_C0(f,t);
        t0 = F_C0_THREAD(f,t);
        F_AREA(A, f, t);/* If face lies at domain boundary, use face values; */
        F_CENTROID(x,f,t);
        rat=1.e-4;

        if (fabs(x[0])<L)
        {
                int curnn=((x[0]/L+1.)*TABLESIZE/2.+.5);
                del=.2263*curvy[curnn];
                al=7.873/(curvy[curnn]+1.e-20);
                r=curvy[curnn];
        }
        else
        {
                del=.2263*ras;/*C_UDMI(c0,t0,2);*/
                al=7.873/ras;/*C_UDMI(c0,t0,2);*/
                r=ras;
        }
        /* If face lies IN the domain, use average of adjacent cells. */
        if (BOUNDARY_FACE_THREAD_P(t)) /*Most face values will be available*/
        {
                        flux=F_FLUX(f,t);
        }
        else
        {
```

```
            c1 = F_C1(f,t); /* Get cell on other side of face */
            t1 = F_C1_THREAD(f,t);
            if (i==0)
            {

      /*(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))*/
                  if
(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.25)&&(C
_UDMI(c1,t1,1)<.75)))
                  {
                        NV_DS(dif, =, 0.,-1.,0.,*,1025.);
                        flux=NV_DOT(dif, A);
                  }
                  else
                  {
                        if
(((C_UDMI(c0,t0,1)>2.25)&&(C_UDMI(c0,t0,1)<3.75))||((C_UDMI(c1,t1,1)>2.25)&&(
C_UDMI(c1,t1,1)<3.75)))
                        {
                              NV_DS(dif, =, 0.,0.,0.,*,1025.);
                              flux=NV_DOT(dif, A);
                        }
                        else
                        {

      flux=.5*(C_UDSI_DIFF(c0,t0,i)+C_UDSI_DIFF(c1,t1,i));
                              NV_DS(dif, =, 0.,flux,0.,*,(-al*tanh(-
r*al+al*x[1]+al*del)-al));
                              flux=F_FLUX(f,t)+NV_DOT(dif, A);
                        }
                  }
            }
            else
            {

      if(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))
                  {
                        NV_DS(dif, =,0.,-1.,0.,*,1025.);
                        flux=NV_DOT(dif, A);
                  }
                  else
                  {
```

```
                        if
(((C_UDMI(c0,t0,1)>2.25)&&(C_UDMI(c0,t0,1)<3.75))||((C_UDMI(c1,t1,1)>2.25)&&(
C_UDMI(c1,t1,1)<3.75)))
                        {
                                NV_DS(dif, =, 0.,0.,0.,*,1025.);
                                flux=NV_DOT(dif, A);
                                if
((C_UDMI(c0,t0,1)>1.25)&&(C_UDMI(c0,t0,1)<2.75))
                                {
                                        if (C_UDSI(c1,t1,1)>0)
                                        {
                                        C_CENTROID(x1,c0,t0);
                                        NV_VV(x2,=,x,-,x1);
                                        NV_VS(x1,=,x2,/,NV_MAG(x2));

        /*flux=1025.*(rat)*(C_UDSI(c0,t0,1)+NV_DOT(C_UDSI_G(c0,t0,1),x2));*/
                                        if (NV_DOT(A,x1)>0)

        flux=1025.*(rat)*(C_UDSI(c0,t0,1)+NV_DOT(C_UDSI_G(c0,t0,1),x2))*NV_M
AG(A);
                                        else
                                                flux=-
1025.*(rat)*(C_UDSI(c0,t0,1)+NV_DOT(C_UDSI_G(c0,t0,1),x2))*NV_MAG(A);

                                        C_UDMI(c0,t0,4)=1025.*(rat);
                                        flux=F_FLUX(f,t)+flux;
                                        }
                                }
                                if
((C_UDMI(c1,t1,1)>1.25)&&(C_UDMI(c1,t1,1)<2.75))
                                {
                                        C_CENTROID(x1,c1,t1);
                                        NV_VV(x2,=,x,-,x1);
                                        NV_VS(x1,=,x2,/,NV_MAG(x2));

        /*flux=1025.*(rat)*(C_UDSI(c1,t1,1)+NV_DOT(C_UDSI_G(c1,t1,1),x2));*/
                                        if (NV_DOT(A,x1)>0)

        flux=1025.*(rat)*(C_UDSI(c1,t1,1)+NV_DOT(C_UDSI_G(c1,t1,1),x2))*NV_M
AG(A);
                                        else
                                                flux=-
1025.*(rat)*(C_UDSI(c1,t1,1)+NV_DOT(C_UDSI_G(c1,t1,1),x2))*NV_MAG(A);

                                        C_UDMI(c1,t1,4)=1025.*(rat);
                                        flux=F_FLUX(f,t)+flux;
```

```
                                        }
                                    }
                                    else
                                    {

        flux=.5*(C_UDSI_DIFF(c0,t0,1)/(C_UDSI(c0,t0,0)+1.e-
15)+C_UDSI_DIFF(c1,t1,1)/(C_UDSI(c1,t1,0)+1.e-15));
                                            NV_DS(dif, =,
.5*(C_UDSI_G(c0,t0,0)[0]+C_UDSI_G(c1,t1,0)[0]),.5*(C_UDSI_G(c0,t0,0)[1]+C_UDS
I_G(c1,t1,0)[1]),0.,*,(-1.)*flux);
                                            flux=F_FLUX(f,t)+NV_DOT(dif, A);
                                    }
                                }
                            }
        }
        return flux;
}
DEFINE_UDS_FLUX(flux_work4,f,t,i)
{
        cell_t c0, c1 = -1;
        Thread *t0, *t1 = NULL;
        real NV_VEC(psi_vec), NV_VEC(A), NV_VEC(dif), flux = 0.0,
x[ND_ND],x1[ND_ND],x2[ND_ND], r, rd;
        real al, del,R, rat;
        c0 = F_C0(f,t);
        t0 = F_C0_THREAD(f,t);
        F_AREA(A, f, t);/* If face lies at domain boundary, use face values; */
        F_CENTROID(x,f,t);
        rat=5.e-5;

        if (fabs(x[0])<L)
        {
                int curnn=((x[0]/L+1.)*TABLESIZE/2.+.5);
                del=.2263*curvy[curnn];
                al=7.873/(curvy[curnn]+1.e-20);
                r=curvy[curnn];
        }
        else
        {
                del=.2263*ras;/*C_UDMI(c0,t0,2);*/
                al=7.873/ras;/*C_UDMI(c0,t0,2);*/
                r=ras;
        }
        /* If face lies IN the domain, use average of adjacent cells. */
        if (BOUNDARY_FACE_THREAD_P(t)) /*Most face values will be available*/
        {
```

```
                flux=F_FLUX(f,t);
        }
        else
        {
                c1 = F_C1(f,t); /* Get cell on other side of face */
                t1 = F_C1_THREAD(f,t);
                if (i==0)
                {

        /*(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))*/
                        if
(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.25)&&(C
_UDMI(c1,t1,1)<.75)))
                        {
                                NV_DS(dif, =, 0.,-1.,0.,*,1025.);
                                flux=NV_DOT(dif, A);
                        }
                        else
                        {
                                if
(((C_UDMI(c0,t0,1)>2.25)&&(C_UDMI(c0,t0,1)<3.75))||((C_UDMI(c1,t1,1)>2.25)&&(
C_UDMI(c1,t1,1)<3.75)))
                                {
                                        NV_DS(dif, =, 0.,0.,0.,*,1025.);
                                        flux=NV_DOT(dif, A);
                                }
                                else
                                {

        flux=.5*(C_UDSI_DIFF(c0,t0,i)+C_UDSI_DIFF(c1,t1,i));
                                        NV_DS(dif, =, 0.,flux,0.,*,(-al*tanh(-
r*al+al*x[1]+al*del)-al));
                                        flux=F_FLUX(f,t)+NV_DOT(dif, A);
                                }
                        }
                }
                else
                {

        if(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))
                        {
                                NV_DS(dif, =,0.,-1.,0.,*,1025.);
                                flux=NV_DOT(dif, A);
                        }
```

```
                else
                {
                        {

        flux=.5*(C_UDSI_DIFF(c0,t0,0)+C_UDSI_DIFF(c1,t1,0));
                                NV_DS(dif, =,
.5*(C_UDSI_G(c0,t0,0)[0]+C_UDSI_G(c1,t1,0)[0]),.5*(C_UDSI_G(c0,t0,0)[1]+C_UDS
I_G(c1,t1,0)[1]),0.,*,(-1.)*flux);
                                flux=F_FLUX(f,t)+NV_DOT(dif, A);
                        }
                }
        }
        }
        return flux;
}
DEFINE_EXECUTE_AT_END(ex_sorce)
{
        Domain *d;
        Thread *t, *t0, *t1, *t2;
        face_t f1;
        cell_t c, c0, c1;
        real A[ND_ND],x[ND_ND],x1[ND_ND],x2[ND_ND], cen, cen2, con, conn,
curv;
        real curvyn[596];
        real curvyt[596];
        int n, loopl, cur;
        d = Get_Domain(1); /* mixture domain if multiphase */
        for (loopl=0;loopl<(TABLESIZE+1);loopl++)
        {
                curvyt[loopl]=0.;
                curvyn[loopl]=0.;
        }
        thread_loop_c(t,d)/*loop through cell
thread*//*&&(CURRENT_TIME>(718.*pow(C_STRAIN_RATE_MAG(c,t)+1.e-20,-
.2)))*/
        {
                if (FLUID_THREAD_P(t))
                {
                        begin_c_loop(c,t)
                        {

                                if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                                {
                                        con=0.;
                                        conn=0;
                                        C_CENTROID(x1,c,t);
```

```
c_face_loop(c, t, n)
{
        f1=C_FACE(c,t,n);
        t1=C_FACE_THREAD(c,t,n);
        F_AREA(A,f1,t1);
        F_CENTROID(x2,f1,t1);
        if (!BOUNDARY_FACE_THREAD_P(t1))
        {
                c1=F_C1(f1,t1);
                t2=THREAD_T1(t1);
                if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                {

        con=NV_MAG(A)*C_UDMI(c1,t2,4)*.5*(C_UDSI(c,t,1)+C_UDSI(c1,t2,1))+co
n;
                        NV_VV(x2,=,x2,-,x1);

        NV_VS(x2,=,x2,/,NV_MAG(x2));
                        if
(NV_DOT(C_UDSI_G(c,t,1),x2)<0)

        conn=(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C_UDSI(c,t,1)*NV_DO
T(C_UDSI_G(c,t,0),x2))*NV_MAG(A)+conn;
                }
                c1=F_C0(f1,t1);
                t2=THREAD_T0(t1);
                if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                {

        con=NV_MAG(A)*C_UDMI(c1,t2,4)*.5*(C_UDSI(c,t,1)+C_UDSI(c1,t2,1))+co
n;
                        NV_VV(x2,=,x2,-,x1);

        NV_VS(x2,=,x2,/,NV_MAG(x2));
                        if
(NV_DOT(C_UDSI_G(c,t,1),x2)<0)

        conn=(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C_UDSI(c,t,1)*NV_DO
T(C_UDSI_G(c,t,0),x2))*NV_MAG(A)+conn;
                }
        }
}
con=con/C_VOLUME(c,t)/1025.;
```

```
                                        conn=-
C_UDSI_DIFF(c,t,1)*conn/C_VOLUME(c,t)/1025./(C_UDSI(c,t,0)+.0000000001);
                                        if (con<conn)
                                                C_UDMI(c,t,3)=con;
                                        else
                                                C_UDMI(c,t,3)=conn;

                                }
                                else
                                {
                                        C_UDMI(c,t,3)=0.;
                                }
                                C_CENTROID(x,c,t);
                                if (fabs(x[0])<L)
                                {
                                        cur=((x[0]/L+1.)*TABLESIZE/2.+.5);
                                        /*C_UDMI(c,t,2)=curvy[cur];*/
                                }
                                /*if ((C_UDMI(c,t,2)>x[1])&&(C_UDMI(c,t,1)<.2))
                                        C_UDMI(c,t,5)=pow((ras-
C_UDMI(c,t,2))/ras,4)*C_VOLUME(c,t)*2.*M_PI/(5.e-14);
                                else
                                        C_UDMI(c,t,5)=0.;     */
                                if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                                {
                                        if (fabs(x[0])<L)
                                        {
                                                curvyt[cur]=curvyt[cur]+x[1];
                                                curvyn[cur]=curvyn[cur]+1.;
                                        }
                                        if
((C_UDMI(c,t,0)>=0.2))/*&&(C_STRAIN_RATE_MAG(c,t)>1000.))
&&(CURRENT_TIME>(718.*pow(C_STRAIN_RATE_MAG(c,t)+1.e-20,-.2)))*/
                                        {
                                                C_UDMI(c,t,0)=C_UDMI(c,t,0)-
C_UDMI(c,t,3)*CURRENT_TIMESTEP;
                                                if (C_UDMI(c,t,0)<=.2)
                                                {
                                                        C_UDMI(c,t,0)=0.;
                                                        C_UDMI(c,t,1)=3.;

        C_UDMI(c,t,6)=(C_VOLUME(c,t)*2.*M_PI);

        C_UDMI(c,t,8)=C_STRAIN_RATE_MAG(c,t);
                                                        C_UDMI(c,t,9)=CURRENT_TIME;
                                                        cen=0.;
```

```
                                                cen2=1.e20;
                                                c_face_loop(c, t, n)
                                                {
                                                        f1=C_FACE(c,t,n);
        t1=C_FACE_THREAD(c,t,n);
                                                        F_CENTROID(x,f1,t1);
                                                        if (x[1]>cen)
                                                                cen=x[1];
                                                        if (x[1]<cen2)
                                                                cen2=x[1];
                                                        if ((100*(ras-
x[1])/ras)>stmin)
                                                                stmin=(100*(ras-
x[1])/ras);
                                                        if
(!BOUNDARY_FACE_THREAD_P(t1))
                                                        {
                                                                c0=F_C1(f1,t1);
                                                                t0=THREAD_T1(t1);
                                                                if
(C_UDMI(c0,t0,1)<.02)
                                                                {
        C_UDMI(c0,t0,1)=2;
        C_UDMI(c0,t0,10)=CURRENT_TIME;
                                                                }
                                                                c0=F_C0(f1,t1);
                                                                t0=THREAD_T0(t1);
                                                                if
(C_UDMI(c0,t0,1)<.02)
                                                                {
        C_UDMI(c0,t0,1)=2;
        C_UDMI(c0,t0,10)=CURRENT_TIME;
                                                                }
                                                        }
                                                }
                                                C_UDMI(c,t,7)=(cen-
cen2)/(CURRENT_TIME-C_UDMI(c,t,10));
                                                C_U(c,t)=0;
                                                C_V(c,t)=0;
                                        }
                                }
```

```
                        }
                        if ((C_UDMI(c,t,1)>2.5)&&(C_UDMI(c,t,1)<3.5))
                        {
                                con=0;
                                c_face_loop(c, t, n)
                                {
                                        f1=C_FACE(c,t,n);
                                        t1=C_FACE_THREAD(c,t,n);
                                        if (!BOUNDARY_FACE_THREAD_P(t1))
                                        {
                                                c1=F_C1(f1,t1);
                                                t2=THREAD_T1(t1);
                                                if
((C_UDMI(c1,t2,1)>1.5)&&(C_UDMI(c1,t2,1)<2.5))
                                                {
                                                        con=1;;
                                                }
                                                c1=F_C0(f1,t1);
                                                t2=THREAD_T0(t1);
                                                if
((C_UDMI(c1,t2,1)>1.5)&&(C_UDMI(c1,t2,1)<2.5))
                                                {
                                                        con=1.;
                                                }
                                        }
                                }
                                if (con<.5)
                                {
                                        C_UDMI(c,t,1)=.45;
                                        C_UDSI(c,t,0)=0;
                                        C_UDSI(c,t,1)=0;
                                }
                        }
                }
                end_c_loop(c,t)
        }
}
for (loopl=0;loopl<(TABLESIZE+1);loopl++)
{
        curv=(ras)-(ras)*st*.5*(1-cos(M_PI*(((L)*(2.*loopl/TABLESIZE-
1.))/(L)+1.)));

        if ((curvyt[loopl]>.0)&&(curvyt[loopl]/(curvyn[loopl]+.000001)<curv))
        {

                curvy[loopl]=curvyt[loopl]/(curvyn[loopl]+.000001);
```

```
                }
                else
                        curvy[loopl]=curv;
                Message ("x: %e",curvy[loopl]);

        }
}

DEFINE_EXECUTE_AT_END(exup_sorce)
{
        Domain *d;
        Thread *t, *t0, *t1, *t2;
        face_t f1;
        cell_t c, c0, c1;
        real A[ND_ND],x[ND_ND],x1[ND_ND],x2[ND_ND], cen, cen2, con, conn,
curv;
        real curvyn[596];
        real curvyt[596];
        int n, loopl, cur;
        d = Get_Domain(1); /* mixture domain if multiphase */
        for (loopl=0;loopl<(TABLESIZE+1);loopl++)
        {
                curvyt[loopl]=0.;
                curvyn[loopl]=0.;
        }
        thread_loop_c(t,d)/*loop through cell
thread*//*&&(CURRENT_TIME>(718.*pow(C_STRAIN_RATE_MAG(c,t)+1.e-20,-
.2)))*/
        {
                if (FLUID_THREAD_P(t))
                {
                        begin_c_loop(c,t)
                        {

                                if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                                {
                                        con=0.;
                                        conn=0.;
                                        cen=0.;
                                        C_CENTROID(x1,c,t);
                                        c_face_loop(c, t, n)
                                        {
                                                f1=C_FACE(c,t,n);
                                                t1=C_FACE_THREAD(c,t,n);
                                                F_AREA(A,f1,t1);
                                                F_CENTROID(x2,f1,t1);
```

```
                                        if (!BOUNDARY_FACE_THREAD_P(t1))
                                        {
                                                if (x2[1]>cen)
                                                {
                                                        cen=x2[1];
                                                        c1=F_C1(f1,t1);
                                                        t2=THREAD_T1(t1);
                                                        if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                                                        {
                                                                NV_VV(x2,=,x2,-
,x1);

        NV_VS(x2,=,x2,/,NV_MAG(x2));
                                                                if
(NV_DOT(C_UDSI_G(c,t,1),x2)<0)

        conn=(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C_UDSI(c,t,1)*NV_DO
T(C_UDSI_G(c,t,0),x2))*NV_MAG(A);
                                                        }
                                                        c1=F_C0(f1,t1);
                                                        t2=THREAD_T0(t1);
                                                        if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                                                        {
                                                                NV_VV(x2,=,x2,-
,x1);

        NV_VS(x2,=,x2,/,NV_MAG(x2));
                                                                if
(NV_DOT(C_UDSI_G(c,t,1),x2)<0)

        conn=(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C_UDSI(c,t,1)*NV_DO
T(C_UDSI_G(c,t,0),x2))*NV_MAG(A);
                                                        }
                                                }
                                        }
                                        con=C_UDMI(c,t,5)/C_VOLUME(c,t)/1025.;
                                        conn=-
C_UDSI_DIFF(c,t,1)*conn/C_VOLUME(c,t)/1025./(C_UDSI(c,t,0)+.0000000001);
                                        if (con<conn)
                                                C_UDMI(c,t,3)=con;
                                        else
                                                C_UDMI(c,t,3)=conn;
```

```
                        }
                        else
                        {
                                C_UDMI(c,t,3)=0.;
                        }
                        C_CENTROID(x,c,t);
                        if (fabs(x[0])<L)
                        {
                                cur=((x[0]/L+1.)*TABLESIZE/2.+.5);
                                /*C_UDMI(c,t,2)=curvy[cur];*/
                        }
                        /*if ((C_UDMI(c,t,2)>x[1])&&(C_UDMI(c,t,1)<.2))
                                C_UDMI(c,t,5)=pow((ras-
C_UDMI(c,t,2))/ras,4)*C_VOLUME(c,t)*2.*M_PI/(5.e-14);
                        else
                                C_UDMI(c,t,5)=0.;     */
                        if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                        {
                                if (fabs(x[0])<L)
                                {
                                        curvyt[cur]=curvyt[cur]+x[1];
                                        curvyn[cur]=curvyn[cur]+1.;
                                }
                                if
((C_UDMI(c,t,0)>=0.2))/*&&(C_STRAIN_RATE_MAG(c,t)>1000.))
&&(CURRENT_TIME>(718.*pow(C_STRAIN_RATE_MAG(c,t)+1.e-20,-.2)))*/
                                {
                                        C_UDMI(c,t,0)=C_UDMI(c,t,0)-
C_UDMI(c,t,3)*CURRENT_TIMESTEP;
                                        if (C_UDMI(c,t,0)<=.2)
                                        {
                                                C_UDMI(c,t,0)=0.;
                                                C_UDMI(c,t,1)=3.;

        C_UDMI(c,t,6)=(C_VOLUME(c,t)*2.*M_PI);

        C_UDMI(c,t,8)=C_STRAIN_RATE_MAG(c,t);
                                                C_UDMI(c,t,9)=CURRENT_TIME;
                                                cen=0.;
                                                cen2=1.e20;
                                                c_face_loop(c, t, n)
                                                {
                                                        f1=C_FACE(c,t,n);

        t1=C_FACE_THREAD(c,t,n);
                                                        F_CENTROID(x,f1,t1);
```

```
                                                    if (x[1]>cen)
                                                            cen=x[1];
                                                    if (x[1]<cen2)
                                                            cen2=x[1];
                                                    if ((100*(ras-
x[1])/ras)>stmin)
                                                            stmin=(100*(ras-
x[1])/ras);
                                                    if
(!BOUNDARY_FACE_THREAD_P(t1))
                                                    {
                                                            c0=F_C1(f1,t1);
                                                            t0=THREAD_T1(t1);
                                                            if
(C_UDMI(c0,t0,1)<.02)
                                                            {

        C_UDMI(c0,t0,1)=2;

        C_UDMI(c0,t0,10)=CURRENT_TIME;
                                                            }
                                                            c0=F_C0(f1,t1);
                                                            t0=THREAD_T0(t1);
                                                            if
(C_UDMI(c0,t0,1)<.02)
                                                            {

        C_UDMI(c0,t0,1)=2;

        C_UDMI(c0,t0,10)=CURRENT_TIME;
                                                            }
                                                    }
                                            }
                                            C_UDMI(c,t,7)=(cen-
cen2)/(CURRENT_TIME-C_UDMI(c,t,10));
                                            C_U(c,t)=0;
                                            C_V(c,t)=0;
                                    }
                            }
                    }
                    if ((C_UDMI(c,t,1)>2.5)&&(C_UDMI(c,t,1)<3.5))
                    {
                            con=0;
                            c_face_loop(c, t, n)
                            {
                                    f1=C_FACE(c,t,n);
```

```
                                        t1=C_FACE_THREAD(c,t,n);
                                        if (!BOUNDARY_FACE_THREAD_P(t1))
                                        {
                                                c1=F_C1(f1,t1);
                                                t2=THREAD_T1(t1);
                                                if
((C_UDMI(c1,t2,1)>1.5)&&(C_UDMI(c1,t2,1)<2.5))
                                                {
                                                        con=1;;
                                                }
                                                c1=F_C0(f1,t1);
                                                t2=THREAD_T0(t1);
                                                if
((C_UDMI(c1,t2,1)>1.5)&&(C_UDMI(c1,t2,1)<2.5))
                                                {
                                                        con=1.;
                                                }
                                        }
                                }
                                if (con<.5)
                                {
                                        C_UDMI(c,t,1)=.45;
                                        C_UDSI(c,t,0)=0;
                                        C_UDSI(c,t,1)=0;
                                }
                        }
                }
                end_c_loop(c,t)
        }
}
for (loopl=0;loopl<(TABLESIZE+1);loopl++)
{
        curv=(ras)-(ras)*st*.5*(1-cos(M_PI*(((L)*(2.*loopl/TABLESIZE-
1.))/(L)+1.)));

        if ((curvyt[loopl]>.0)&&(curvyt[loopl]/(curvyn[loopl]+.000001)<curv))
        {

                curvy[loopl]=curvyt[loopl]/(curvyn[loopl]+.000001);
        }
        else
                curvy[loopl]=curv;
        Message ("x: %e",curvy[loopl]);

}
}
```

```
DEFINE_EXECUTE_AT_END(exup2_sorce)
{
        Domain *d;
        Thread *t, *t0, *t1, *t2;
        face_t f1;
        cell_t c, c0, c1;
        real A[ND_ND],x[ND_ND],x1[ND_ND],x2[ND_ND],x3[ND_ND], cen, cen2,
con, conn, curv, col;
        real curvyn[596];
        real curvyt[596];
        real smth[600];
        int n, nl, loo, lo, loopl, cur;
        d = Get_Domain(1); /* mixture domain if multiphase */
        for (loopl=0;loopl<(TABLESIZE+1);loopl++)
        {
                curvyt[loopl]=0.;
                curvyn[loopl]=0.;
        }
        thread_loop_c(t,d)/*loop through cell
thread*//*&&(CURRENT_TIME>(718.*pow(C_STRAIN_RATE_MAG(c,t)+1.e-20,-
.2)))*/
        {
                if (FLUID_THREAD_P(t))
                {
                        begin_c_loop(c,t)
                        {

                                if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                                {
                                        con=0.;
                                        conn=0.;
                                        cen=0.;
                                        C_CENTROID(x1,c,t);
                                        c_face_loop(c, t, n)
                                        {

                                                f1=C_FACE(c,t,n);
                                                t1=C_FACE_THREAD(c,t,n);
                                                F_AREA(A,f1,t1);
                                                F_CENTROID(x2,f1,t1);
                                                if (x2[1]>cen)
                                        {
                                                cen=x2[1];
                                                if
(!BOUNDARY_FACE_THREAD_P(t1))
```

```
                                                {
                                                        c1=F_C1(f1,t1);
                                                        t2=THREAD_T1(t1);
                                                        if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                                                        {
                                                                NV_VV(x,=,x2,-,x1);

        NV_VS(x3,=,x,/,NV_MAG(x));

        con=C_UDMI(c,t,4)*(C_UDSI(c,t,1)+NV_DOT(C_UDSI_G(c,t,1),x))*NV_MA
G(A);
                                                                if
(NV_DOT(C_UDSI_G(c,t,1),x3)<0)
                                                                {

        conn=(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x3)+C_UDSI(c,t,1)*NV_DO
T(C_UDSI_G(c,t,0),x3))*NV_MAG(A);
                                                                }

                                                        }
                                                        c1=F_C0(f1,t1);
                                                        t2=THREAD_T0(t1);
                                                        if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                                                        {
                                                                NV_VV(x,=,x2,-,x1);

        NV_VS(x3,=,x,/,NV_MAG(x));

        con=C_UDMI(c,t,4)*(C_UDSI(c,t,1)+NV_DOT(C_UDSI_G(c,t,1),x))*NV_MA
G(A);
                                                                if
(NV_DOT(C_UDSI_G(c,t,1),x3)<0)
                                                                {

        conn=(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x3)+C_UDSI(c,t,1)*NV_DO
T(C_UDSI_G(c,t,0),x3))*NV_MAG(A);
                                                                }

                                                        }
                                                }
                                        }
                                }
                        }
                        con=con/C_VOLUME(c,t)/1025.;
```

```
                                        conn=-
C_UDSI_DIFF(c,t,1)*conn/C_VOLUME(c,t)/1025./(C_UDSI(c,t,0)+.0000000001);
                                        if (con<conn)
                                                C_UDMI(c,t,3)=con;
                                        else
                                                C_UDMI(c,t,3)=conn;
                                        cur=((x1[0]/L+1.)*599/2.+.5);
                                        smth[cur]=C_UDMI(c,t,3);

                                }
                                else
                                {
                                        C_UDMI(c,t,3)=0.;
                                }
                        }
                        end_c_loop(c,t)
for (loo=0;loo<(10);loo++)
                                {
                                for (loopl=5;loopl<(595);loopl++)
                                {
                                        nl=0;
                                        col=0.;
                                        for (lo=-4;lo<5;lo++)
                                        {
                                                if (smth[loopl+lo]>0.)
                                                {
                                                        col=col+smth[loopl+lo];
                                                        nl=nl+1;
                                                }
                                        }
                                        smth[loopl]=col/(nl+1.e-20);
                                }}
                                begin_c_loop(c,t)
                        {
                                C_CENTROID(x,c,t);
                                if (fabs(x[0])<L)
                                {
                                        cur=((x[0]/L+1.)*TABLESIZE/2.+.5);
                                        /*C_UDMI(c,t,2)=curvy[cur];*/
                                }
                                /*if ((C_UDMI(c,t,2)>x[1])&&(C_UDMI(c,t,1)<.2))
                                        C_UDMI(c,t,5)=pow((ras-
C_UDMI(c,t,2))/ras,4)*C_VOLUME(c,t)*2.*M_PI/(5.e-14);
                                else
                                        C_UDMI(c,t,5)=0.;    */
                                if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
```

```
                {
                        if (fabs(x[0])<L)
                        {
                                curvyt[cur]=curvyt[cur]+x[1];
                                curvyn[cur]=curvyn[cur]+1.;
                        }
                        cur=((x[0]/L+1.)*599/2.+.5);
                        if
((C_UDMI(c,t,0)>=0.2))/*&&(C_STRAIN_RATE_MAG(c,t)>1000.))
&&(CURRENT_TIME>(718.*pow(C_STRAIN_RATE_MAG(c,t)+1.e-20,-.2)))*/
                        {
                                C_UDMI(c,t,0)=C_UDMI(c,t,0)-
smth[cur]*CURRENT_TIMESTEP;
                                if (C_UDMI(c,t,0)<=.2)
                                {
                                        C_UDMI(c,t,0)=0.;
                                        C_UDMI(c,t,1)=3.;

        C_UDMI(c,t,6)=(C_VOLUME(c,t)*2.*M_PI);

        C_UDMI(c,t,8)=C_STRAIN_RATE_MAG(c,t);
                                        C_UDMI(c,t,9)=CURRENT_TIME;
                                        cen=0.;
                                        cen2=1.e20;
                                        c_face_loop(c, t, n)
                                        {
                                                f1=C_FACE(c,t,n);

        t1=C_FACE_THREAD(c,t,n);
                                                F_CENTROID(x,f1,t1);
                                                if (x[1]>cen)
                                                        cen=x[1];
                                                if (x[1]<cen2)
                                                        cen2=x[1];
                                                if ((100*(ras-
x[1])/ras)>stmin)
                                                        stmin=(100*(ras-
x[1])/ras);
                                                if
(!BOUNDARY_FACE_THREAD_P(t1))
                                                {
                                                        c0=F_C1(f1,t1);
                                                        t0=THREAD_T1(t1);
                                                        if
(C_UDMI(c0,t0,1)<.02)
                                                        {
```

```
        C_UDMI(c0,t0,1)=2;

        C_UDMI(c0,t0,10)=CURRENT_TIME;
                                                                }
                                                                c0=F_C0(f1,t1);
                                                                t0=THREAD_T0(t1);
                                                                if
(C_UDMI(c0,t0,1)<.02)
                                                                {

        C_UDMI(c0,t0,1)=2;

        C_UDMI(c0,t0,10)=CURRENT_TIME;
                                                                }
                                                        }
                                                }
                                                C_UDMI(c,t,7)=(cen-
cen2)/(CURRENT_TIME-C_UDMI(c,t,10)+.00001);
                                                C_U(c,t)=0;
                                                C_V(c,t)=0;
                                        }
                                }
                        }
                        if ((C_UDMI(c,t,1)>2.5)&&(C_UDMI(c,t,1)<3.5))
                        {
                                con=0;
                                c_face_loop(c, t, n)
                                {
                                        f1=C_FACE(c,t,n);
                                        t1=C_FACE_THREAD(c,t,n);
                                        if (!BOUNDARY_FACE_THREAD_P(t1))
                                        {
                                                c1=F_C1(f1,t1);
                                                t2=THREAD_T1(t1);
                                                if
((C_UDMI(c1,t2,1)>1.5)&&(C_UDMI(c1,t2,1)<2.5))
                                                {
                                                        con=1;;
                                                }
                                                c1=F_C0(f1,t1);
                                                t2=THREAD_T0(t1);
                                                if
((C_UDMI(c1,t2,1)>1.5)&&(C_UDMI(c1,t2,1)<2.5))
                                                {
                                                        con=1.;
```

```
                                                }
                                        }
                                }
                                if (con<.5)
                                {
                                        C_UDMI(c,t,1)=.45;
                                        C_UDSI(c,t,0)=0;
                                        C_UDSI(c,t,1)=0;
                                }
                        }
                }
                end_c_loop(c,t)
        }
    }
    for (loopl=0;loopl<(TABLESIZE+1);loopl++)
    {
        curv=(ras)-(ras)*st*.5*(1-cos(M_PI*(((L)*(2.*loopl/TABLESIZE-
1.))/(L)+1.)));

        if ((curvyt[loopl]>.0)&&(curvyt[loopl]/(curvyn[loopl]+.000001)<curv))
        {

            curvy[loopl]=curvyt[loopl]/(curvyn[loopl]+.000001);
        }
        else
            curvy[loopl]=curv;
        Message ("x: %e",curvy[loopl]);

    }
}

DEFINE_ON_DEMAND(calc)
{
    Domain *d;

Thread *t, *t0, *t1, *t2;
    face_t f1;
    cell_t c, c0, c1;
    real A[ND_ND],x[ND_ND],x1[ND_ND],x2[ND_ND], cen, cen2, con, conn,
curv;

    real curvyn[596];
    real curvyt[596];
    int n, loopl, cur;
    d = Get_Domain(1); /* mixture domain if multiphase */
```

```
        thread_loop_c(t,d)/*loop through cell
thread*//*&&(CURRENT_TIME>(718.*pow(C_STRAIN_RATE_MAG(c,t)+1.e-20,-
.2)))*/
        {
                if (FLUID_THREAD_P(t))
                {
                        begin_c_loop(c,t)
                        {
                                if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                                {
                                        con=0.;
                                        conn=0;
                                        C_CENTROID(x1,c,t);
                                        c_face_loop(c, t, n)
                                        {
                                                f1=C_FACE(c,t,n);
                                                t1=C_FACE_THREAD(c,t,n);
                                                F_AREA(A,f1,t1);
                                                F_CENTROID(x2,f1,t1);
                                                if (!BOUNDARY_FACE_THREAD_P(t1))
                                                {
                                                        c1=F_C1(f1,t1);
                                                        t2=THREAD_T1(t1);
                                                        if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                                                        {

        con=NV_MAG(A)*C_UDMI(c1,t2,4)*.5*(C_UDSI(c,t,1)+C_UDSI(c1,t2,1))+co
n;
                                                                NV_VV(x2,=,x2,-,x1);

        NV_VS(x2,=,x2,/,NV_MAG(x2));
                                                                if
(NV_DOT(C_UDSI_G(c,t,1),x2)<0)

        conn=NV_DOT(C_UDSI_G(c,t,1),x2)*NV_MAG(A)+conn;
                                                        }
                                                        c1=F_C0(f1,t1);
                                                        t2=THREAD_T0(t1);
                                                        if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                                                        {

        con=NV_MAG(A)*C_UDMI(c1,t2,4)*.5*(C_UDSI(c,t,1)+C_UDSI(c1,t2,1))+co
n;
                                                                NV_VV(x2,=,x2,-,x1);
```

```
        NV_VS(x2,=,x2,/,NV_MAG(x2));
                                                        if
(NV_DOT(C_UDSI_G(c,t,1),x2)<0)

        conn=NV_DOT(C_UDSI_G(c,t,1),x2)*NV_MAG(A)+conn;
                                                }
                                        }
                                }
                                con=con/C_VOLUME(c,t)/1025.;
                                conn=-
C_UDSI_DIFF(c,t,0)*conn/C_VOLUME(c,t)/1025.;
                                if (con<conn)
                                        C_UDMI(c,t,3)=con;
                                else
                                        C_UDMI(c,t,3)=conn;

                        }
                        else
                        {
                                C_UDMI(c,t,3)=0.;
                        }
                        C_CENTROID(x,c,t);
                        if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                        {
                                cen=0.;
                                cen2=1.e20;
                                c_face_loop(c, t, n)
                                {
                                        f1=C_FACE(c,t,n);
                                        t1=C_FACE_THREAD(c,t,n);
                                        F_CENTROID(x,f1,t1);
                                        if (x[1]>cen)
                                                cen=x[1];
                                        if (x[1]<cen2)
                                                cen2=x[1];
                                }
                                C_UDMI(c,t,2)=(60.e6)*(cen-
cen2)*C_UDMI(c,t,3);

                        }
                        else
                                C_UDMI(c,t,2)=0.;
                }
                end_c_loop(c,t)
        }
```

```
        }

}

DEFINE_ON_DEMAND(calc_flx)
{
        Domain *d;

Thread *t, *t0, *t1, *t2;
        face_t f1;
        cell_t c, c0, c1;
        real A[ND_ND],x[ND_ND],x1[ND_ND],x2[ND_ND], cen, cen2, con, conn,
curv;

        real curvyn[596];
        real curvyt[596];
        int n, loopl, cur;
        d = Get_Domain(1); /* mixture domain if multiphase */

        thread_loop_c(t,d)/*loop through cell
thread*//*&&(CURRENT_TIME>(718.*pow(C_STRAIN_RATE_MAG(c,t)+1.e-20,-
.2)))*/
        {
                if (FLUID_THREAD_P(t))
                {
                        begin_c_loop(c,t)
                        {
                                if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                                {
                                        con=0.;
                                        conn=0;
                                        C_CENTROID(x1,c,t);
                                        c_face_loop(c, t, n)
                                        {
                                                f1=C_FACE(c,t,n);
                                                t1=C_FACE_THREAD(c,t,n);
                                                F_AREA(A,f1,t1);
                                                F_CENTROID(x2,f1,t1);
                                                if (!BOUNDARY_FACE_THREAD_P(t1))
                                                {
                                                        c1=F_C1(f1,t1);
                                                        t2=THREAD_T1(t1);
                                                        if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                                                        {
```

```
        con=NV_MAG(A)*C_UDMI(c1,t2,4)*.5*(C_UDSI(c,t,1)+C_UDSI(c1,t2,1))+co
n;
                                                        NV_VV(x2,=,x2,-,x1);

        NV_VS(x2,=,x2,/,NV_MAG(x2));
                                                        if
(NV_DOT(C_UDSI_G(c,t,1),x2)<0)

        conn=(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C_UDSI(c,t,1)*NV_DO
T(C_UDSI_G(c,t,0),x2))*NV_MAG(A)+conn;
                                                }
                                                c1=F_C0(f1,t1);
                                                t2=THREAD_T0(t1);
                                                if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                                                {

        con=NV_MAG(A)*C_UDMI(c1,t2,4)*.5*(C_UDSI(c,t,1)+C_UDSI(c1,t2,1))+co
n;
                                                        NV_VV(x2,=,x2,-,x1);

        NV_VS(x2,=,x2,/,NV_MAG(x2));
                                                        if
(NV_DOT(C_UDSI_G(c,t,1),x2)<0)

        conn=(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C_UDSI(c,t,1)*NV_DO
T(C_UDSI_G(c,t,0),x2))*NV_MAG(A)+conn;
                                                }
                                        }
                                }
                                con=con/C_VOLUME(c,t)/1025.;
                                conn=-
C_UDSI_DIFF(c,t,1)*conn/C_VOLUME(c,t)/1025./(.00000000001+C_UDSI(c,t,0));
                                if (con<conn)
                                        C_UDMI(c,t,3)=con;
                                else
                                        C_UDMI(c,t,3)=conn;

                        }
                        else
                        {
                                C_UDMI(c,t,3)=0.;
                        }
                        C_CENTROID(x,c,t);
                        if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
```

```
                {
                        cen=0.;
                        cen2=1.e20;
                        c_face_loop(c, t, n)
                        {
                                f1=C_FACE(c,t,n);
                                t1=C_FACE_THREAD(c,t,n);
                                F_CENTROID(x,f1,t1);
                                if (x[1]>cen)
                                        cen=x[1];
                                if (x[1]<cen2)
                                        cen2=x[1];
                        }
                        C_UDMI(c,t,2)=(60.e6)*(cen-
cen2)*C_UDMI(c,t,3);

                }
                else
                        C_UDMI(c,t,2)=0.;
            }
            end_c_loop(c,t)
        }
    }

}

DEFINE_ON_DEMAND(calcup_flx)
{
    Domain *d;

Thread *t, *t0, *t1, *t2;
    face_t f1;
    cell_t c, c0, c1;
    real A[ND_ND],x[ND_ND],x1[ND_ND],x2[ND_ND],x3[ND_ND], cen, cen2,
con, conn, curv, col;
    real smth[600];
    real curvyn[596];
    real curvyt[596];
    int n, nl, loo, lo, loopl, cur;
    d = Get_Domain(1); /* mixture domain if multiphase */

    thread_loop_c(t,d)/*loop through cell
thread*//*&&(CURRENT_TIME>(718.*pow(C_STRAIN_RATE_MAG(c,t)+1.e-20,-
.2)))*/
    {
        if (FLUID_THREAD_P(t))
```

```
{
        begin_c_loop(c,t)
        {
                if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                {
                        con=0.;
                        conn=0;
                        cen=0.;
                        C_CENTROID(x1,c,t);
                        c_face_loop(c, t, n)
                        {
                                f1=C_FACE(c,t,n);
                                t1=C_FACE_THREAD(c,t,n);
                                F_AREA(A,f1,t1);
                                F_CENTROID(x2,f1,t1);
                                if (x2[1]>cen)
                                {
                                        cen=x2[1];
                                        if
(!BOUNDARY_FACE_THREAD_P(t1))
                                        {
                                                c1=F_C1(f1,t1);
                                                t2=THREAD_T1(t1);
                                                if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                                                {
                                                        NV_VV(x,=,x2,-,x1);

        NV_VS(x3,=,x,/,NV_MAG(x));

        con=C_UDMI(c,t,4)*(C_UDSI(c,t,1)+NV_DOT(C_UDSI_G(c,t,1),x))*NV_MA
G(A);
                                                        C_UDMI(c,t,5)=2;
                                                        if
(NV_DOT(C_UDSI_G(c,t,1),x3)<0)
                                                        {

        conn=(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x3)+C_UDSI(c,t,1)*NV_DO
T(C_UDSI_G(c,t,0),x3))*NV_MAG(A);

        /*C_UDMI(c,t,10)=((-
60.e6)/1025.)*C_UDSI_DIFF(c,t,1)*(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C
_UDSI(c,t,1)*NV_DOT(C_UDSI_G(c,t,0),x2));*/
                                                        }

        /*con=NV_MAG(A)*C_UDMI(c,t,4)*(C_UDSI(c,t,1));
```

```
        C_UDMI(c,t,10)=((60.e6)/1025.)*C_UDMI(c,t,4)*C_UDSI(c,t,1);
                                                                NV_VV(x2,=,x2,-
,x1);

        NV_VS(x2,=,x2,/,NV_MAG(x2));
                                                                if
(NV_DOT(C_UDSI_G(c,t,1),x2)<0)
                                                                {

        conn=(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C_UDSI(c,t,1)*NV_DO
T(C_UDSI_G(c,t,0),x2))*NV_MAG(A);
                                                                C_UDMI(c,t,10)=((-
60.e6)/1025.)*C_UDSI_DIFF(c,t,1)*(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C
_UDSI(c,t,1)*NV_DOT(C_UDSI_G(c,t,0),x2));
                                                                }*/
                                                        }
                                                        c1=F_C0(f1,t1);
                                                        t2=THREAD_T0(t1);
                                                        if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                                                        {
                                                                NV_VV(x,=,x2,-,x1);

        NV_VS(x3,=,x,/,NV_MAG(x));

        con=C_UDMI(c,t,4)*(C_UDSI(c,t,1)+NV_DOT(C_UDSI_G(c,t,1),x))*NV_MA
G(A);
                                                                C_UDMI(c,t,5)=1;
                                                                if
(NV_DOT(C_UDSI_G(c,t,1),x3)<0)
                                                                {

        conn=(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x3)+C_UDSI(c,t,1)*NV_DO
T(C_UDSI_G(c,t,0),x3))*NV_MAG(A);

        /*C_UDMI(c,t,10)=((-
60.e6)/1025.)*C_UDSI_DIFF(c,t,1)*(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C
_UDSI(c,t,1)*NV_DOT(C_UDSI_G(c,t,0),x2));*/
                                                                }

        /*con=NV_MAG(A)*C_UDMI(c,t,4)*.5*(C_UDSI(c,t,1));

        C_UDMI(c,t,10)=((60.e6)/1025.)*C_UDMI(c,t,4)*C_UDSI(c,t,1);
                                                                NV_VV(x2,=,x2,-
,x1);
```

```
        NV_VS(x2,=,x2,/,NV_MAG(x2));
                                                                if
(NV_DOT(C_UDSI_G(c,t,1),x2)<0)
                                                                {

        conn=(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C_UDSI(c,t,1)*NV_DO
T(C_UDSI_G(c,t,0),x2))*NV_MAG(A);
                                                                C_UDMI(c,t,10)=((-
60.e6)/1025.)*C_UDSI_DIFF(c,t,1)*(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C
_UDSI(c,t,1)*NV_DOT(C_UDSI_G(c,t,0),x2));
                                                                }*/
                                                        }
                                                }
                                        }
                                }
                                con=con/C_VOLUME(c,t)/1025.;
                                conn=-
C_UDSI_DIFF(c,t,1)*conn/C_VOLUME(c,t)/1025./(.00000000001+C_UDSI(c,t,0));

        /*C_UDMI(c,t,3)=C_UDMI(c,t,5)/C_VOLUME(c,t)/1025.;*/
                                if (con<conn)
                                        C_UDMI(c,t,3)=con;
                                else
                                        C_UDMI(c,t,3)=conn;
                                cur=((x1[0]/L+1.)*599/2.+.5);
                                smth[cur]=C_UDMI(c,t,3);

                        }
                        else
                        {
                                C_UDMI(c,t,3)=0.;
                        }
                }
                end_c_loop(c,t)
                        for (loo=0;loo<(10);loo++)
                        {
                        for (loopl=5;loopl<(595);loopl++)
                        {
                                nl=0;
                                col=0.;
                                for (lo=-4;lo<5;lo++)
                                {
                                        if (smth[loopl+lo]>0.)
                                        {
                                                col=col+smth[loopl+lo];
```

```
                                        nl=nl+1;
                                    }
                                }
                                smth[loopl]=col/(nl+1.e-20);
                            }}
                            begin_c_loop(c,t)
                        {
                            C_CENTROID(x,c,t);
                            if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                            {
                                cen=0.;
                                cen2=1.e20;
                                c_face_loop(c, t, n)
                                {
                                    f1=C_FACE(c,t,n);
                                    t1=C_FACE_THREAD(c,t,n);
                                    F_CENTROID(x,f1,t1);
                                    if (x[1]>cen)
                                        cen=x[1];
                                    if (x[1]<cen2)
                                        cen2=x[1];
                                }
                                cur=((x[0]/L+1.)*599/2.+.5);
                                C_UDMI(c,t,2)=(60.e6)*(cen-cen2)*smth[cur]/.8;

                            }
                            else
                                C_UDMI(c,t,2)=0.;
                        }
                        end_c_loop(c,t)
                    }
                }

}


DEFINE_ON_DEMAND(limit_tim)
/*Used to initialize the user defined memory to specific values*/
{
        cell_t c;
        Domain *d;
        Domain *dom;
        Thread *t;
        face_t f, f1, f2, f3, fx, fy;
        Thread *t2, *t1, *t0, *tx, *t3, *ty, *tcy, *tcyy;
        cell_t c0, c1, cy, cyy;
```

```
        real x[ND_ND];
        real a[ND_ND];
        real curv, cen, cen2, cen3, area, st;
        int id, n, goo;
        int loopl;

        d=Get_Domain(1);
        thread_loop_c (t,d)
        {
                begin_c_loop (c,t)
                {
                        if (C_UDMI(c,t,9)>1000.)
                                C_UDMI(c,t,9)=1000.;
                }
                end_c_loop (c,t)
        }
}
DEFINE_ON_DEMAND(mem_8_init)
/*Used to initialize the user defined memory to specific values*/
{
        cell_t c;
        Domain *d;
        Domain *dom;
        Thread *t;
        face_t f, f1, f2, f3, fx, fy;
        Thread *t2, *t1, *t0, *tx, *t3, *ty, *tcy, *tcyy;
        cell_t c0, c1, cy, cyy;
        real x[ND_ND];
        real a[ND_ND];
        real curv, cen, cen2, cen3, area;
        int id, n, goo;
        int loopl;
        stmin=100.*st;
        for (loopl=0;loopl<(TABLESIZE+1);loopl++)
        {
                curvy[loopl]=(ras)-(ras)*st*.5*(1-
cos(M_PI*(((L)*(2.*loopl/TABLESIZE-1.))/(L)+1.)));
                Message ("x: %e",curvy[loopl]);
        }
        d=Get_Domain(1);
        thread_loop_c (t,d)
        {
                begin_c_loop (c,t)
                {
                        C_UDMI(c,t,0)=1.;
                        C_UDMI(c,t,1)=0.;
```

```
                C_UDMI(c,t,2)=ras;
                C_UDMI(c,t,3)=0.;
                C_UDMI(c,t,4)=0.;
                C_UDMI(c,t,5)=0.;
                C_UDMI(c,t,6)=0.;
                C_UDMI(c,t,7)=0.;
                C_UDMI(c,t,8)=0.;
                C_UDMI(c,t,9)=10000.;
                C_UDMI(c,t,10)=0.;
                C_CENTROID(x,c,t);
                if (fabs(x[0])<=L)
                {
                        int cur=((x[0]/L+1.)*TABLESIZE/2.+.5);
                        curv=(ras-ras*st*.5*(1.-cos(M_PI*(x[0]/L+1.))));
                        C_UDMI(c,t,2)=curvy[cur];
                        if (x[1]>curv)
                        {
                                C_UDMI(c,t,1)=.5;
                                C_UDMI(c,t,0)=0.;
                        }
                }
        }
        end_c_loop (c,t)
}
thread_loop_c (t,d)
{
        begin_c_loop (c,t)
        {
                C_CENTROID(x,c,t);
                if (fabs(x[0])<=(L))
                {
                        if (C_UDMI(c,t,1)<.25)
                        {
                                c_face_loop(c, t, n)
                                {
                                        f1=C_FACE(c,t,n);
                                        t1=C_FACE_THREAD(c,t,n);
                                        if (!BOUNDARY_FACE_THREAD_P(t1))
                                        {
                                                c1=F_C1(f1,t1);
                                                t2=THREAD_T1(t1);

if((C_UDMI(c1,t2,1)>.2)&&(C_UDMI(c1,t2,1)<.7))
                                                        C_UDMI(c,t,1)=2.;
                                                c1=F_C0(f1,t1);
                                                t2=THREAD_T0(t1);
```

```
if((C_UDMI(c1,t2,1)>.2)&&(C_UDMI(c1,t2,1)<.7))
                                                C_UDMI(c,t,1)=2.;
                                        }
                                }
                        }
                }
        }
        end_c_loop (c,t)
}
thread_loop_c (t,d)
{
        begin_c_loop (c,t)
        {
                C_CENTROID(x,c,t);
                        if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                        {
                                c_face_loop(c, t, n)
                                {
                                        f1=C_FACE(c,t,n);
                                        t1=C_FACE_THREAD(c,t,n);
                                        if (!BOUNDARY_FACE_THREAD_P(t1))
                                        {
                                                c1=F_C1(f1,t1);
                                                t2=THREAD_T1(t1);

if((C_UDMI(c1,t2,1)>.2)&&(C_UDMI(c1,t2,1)<.7))
                                                        C_UDMI(c1,t2,1)=3.;
                                                c1=F_C0(f1,t1);
                                                t2=THREAD_T0(t1);

if((C_UDMI(c1,t2,1)>.2)&&(C_UDMI(c1,t2,1)<.7))
                                                C_UDMI(c1,t2,1)=3.;
                                        }
                                }
                        }
        }
        end_c_loop (c,t)
}
/*t3 = Lookup_Thread(d,6);
begin_f_loop(f3,t3)
{
        F_CENTROID(x,f3,t3);
        if ((x[0]>(-5*ras))&&(x[0]<(20*ras)))
        {
                c0=F_C0(f3,t3);
```

```
                        t0=THREAD_T0(t3);
                        if ((C_UDMI(c0,t0,1)==0)&&(C_UDMI(c0,t0,0)>.02))
                        {
                                C_UDMI(c0,t0,1)=2;
                        }
                }
        }
        end_f_loop(f3,t3)*/
}
```

Appendix L: C Code for Uniform RBC User-Defined Functions in Fluent to Characterize Species Transport Model (Chapter 4)

```
#include "udf.h"
#include <math.h>
#include "math.h"
const TABLESIZE=595;
real L=2.379e-3;
real ras=.00075;
real curvy[596];
real st=.8484;/*8484*/
real vel=.002358;
real stmin;
/*real vel=.045;*/

#define TSTART 0.0 /* field applied at t = tstart */
/*Define porosity from 0-1 in the computational domain based on user defined memory
(C_UDMI(0))*/


DEFINE_DIFFUSIVITY(const_RBC,c,t,i)
{
        real x[ND_ND], r;
        C_CENTROID(x,c,t);
        if ((C_UDMI(c,t,1)>0.25)&&(C_UDMI(c,t,1)<.75))
                return 0.;
        else
        {
                if (x[0]<-.005)
                        return 1025.*(1.6e-6);
                else
                {
                        return 1025.*(.15*(4.2e-6)*(4.2e-
6)*.45*0.62*C_STRAIN_RATE_MAG(c,t)+1.6e-13);
                }
        }
}
DEFINE_PROFILE(vel_,t,i)
{
        face_t f;
        real x[ND_ND];
        begin_f_loop(f,t)
        {
                F_CENTROID(x,f,t);
```

```
                F_PROFILE(f,t,i)=vel*2.*(1.-pow(x[1]/ras,2));/*14.9626*fourier;*/
        }
        end_f_loop(f,t)
}
DEFINE_PROFILE(out0,t,i)
{
        face_t f;
        cell_t c;
        Thread *t1;
        begin_f_loop(f,t)
        {
                c=F_C0(f,t);
                t1=THREAD_T0(t);
                F_PROFILE(f,t,i)= C_UDSI(c,t1,i);
        }
        end_f_loop(f,t)
}

DEFINE_SOURCE(k1_simp,c,t,dS,eqn)
{
        real cen, con, conn, source;
        real A[ND_ND],x1[ND_ND],x2[ND_ND],mult[ND_ND];
        real k_rat;
        int n;
        face_t f1;
        cell_t c1;
        Thread *t1,*t2;
        k_rat=1025.*(2.e-4-(1.e-9)*C_STRAIN_RATE_MAG(c,t));/*1025.*1.e-5-(1.e-
9)*1025.*C_STRAIN_RATE_MAG(c,t);*/
        k_rat=1025.*(10000.);
        if (k_rat<(1025.*1.e-5))
                k_rat=1025.*1000000.;
        if((C_UDMI(c,t,1)>2.5))/* &&(C_STRAIN_RATE_MAG(c,t)>1000.)
&&(CURRENT_TIME>(718.*pow(C_STRAIN_RATE_MAG(c,t)+1.e-20,-.2))))     */
        {
                if (C_UDSI(c,t,0)<0.)
                {
                        source=0.;
                        dS[eqn] =0.;
                }
                else
                {
                        source=-k_rat*C_UDSI(c,t,0);
                        dS[eqn] =-k_rat;
                }
        }
```

```
        else
        {
                source = 0.;
                dS[eqn] = 0.;
        }
        /*C_UDMI(c,t,4)=k_rat;*/
        return source;
}
DEFINE_PROFILE(vis_res,t,i)
{
        real x[ND_ND];
        real a;
        cell_t c;
        begin_c_loop(c,t)
        {
                C_CENTROID(x,c,t);
                if(((C_UDMI(c,t,1)<.75)&&(C_UDMI(c,t,1)>.25)))
                        a = 1.0e50;
                else
                {
                        if(((C_UDMI(c,t,1)<3.75)&&(C_UDMI(c,t,1)>2.25)))
                                a = 1.0e50;
                        else
                                a = 0.0;
                /*      if ((C_UDMI(c,t,1)<.49)&&(C_UDMI(c,t,1)>.41))
                                a=(CURRENT_TIME-
C_UDMI(c,t,9))*(1e30)/(C_UDMI(c,t,9)-C_UDMI(c,t,10));
                        else
                                a = 0.0;
                        if (a>1.e30)
                                a=1.e30;*/
                }
                F_PROFILE(c,t,i) = a;
        }
        end_c_loop(c,t)
}

DEFINE_UDS_FLUX(flux_work32,f,t,i)
{
        cell_t c0, c1 = -1;
        Thread *t0, *t1 = NULL;
        real NV_VEC(psi_vec), NV_VEC(A), NV_VEC(dif), flux = 0.0,
x[ND_ND],x1[ND_ND],x2[ND_ND], r, rd;
        real al, del,R, rat;
        c0 = F_C0(f,t);
        t0 = F_C0_THREAD(f,t);
```

```
        F_AREA(A, f, t);/* If face lies at domain boundary, use face values; */
        F_CENTROID(x,f,t);
        rat=1.e-1;
        /* If face lies IN the domain, use average of adjacent cells. */
        if (BOUNDARY_FACE_THREAD_P(t)) /*Most face values will be available*/
        {
                        flux=F_FLUX(f,t);
        }
        else
        {
                c1 = F_C1(f,t); /* Get cell on other side of face */
                t1 = F_C1_THREAD(f,t);
                {

        if(((C_UDMI(c0,t0,1)>0.25)&&(C_UDMI(c0,t0,1)<.75))||((C_UDMI(c1,t1,1)>0.
25)&&(C_UDMI(c1,t1,1)<.75)))
                        {
                                NV_DS(dif, =,0.,-1.,0.,*,1025.);
                                flux=NV_DOT(dif, A);
                        }
                        else
                        {
                                if
(((C_UDMI(c0,t0,1)>2.25)&&(C_UDMI(c0,t0,1)<3.75))||((C_UDMI(c1,t1,1)>2.25)&&(
C_UDMI(c1,t1,1)<3.75)))
                                {
                                        NV_DS(dif, =, 0.,0.,0.,*,1025.);
                                        flux=NV_DOT(dif, A);
                                        if
((C_UDMI(c0,t0,1)>1.25)&&(C_UDMI(c0,t0,1)<2.75))
                                        {
                                                C_CENTROID(x1,c0,t0);
                                                NV_VV(x2,=,x,-,x1);
                                                NV_VS(x1,=,x2,/,NV_MAG(x2));

        /*flux=1025.*(rat)*(C_UDSI(c0,t0,1)+NV_DOT(C_UDSI_G(c0,t0,1),x2));*/
                                                if (NV_DOT(A,x1)>0)

        flux=1025.*(rat)*(C_UDSI(c0,t0,0)+NV_DOT(C_UDSI_G(c0,t0,0),x2))*NV_M
AG(A);
                                                else
                                                        flux=-
1025.*(rat)*(C_UDSI(c0,t0,0)+NV_DOT(C_UDSI_G(c0,t0,0),x2))*NV_MAG(A);

                                                C_UDMI(c0,t0,4)=1025.*(rat);
                                                flux=flux;
```

```
                                        }
                                        if
((C_UDMI(c1,t1,1)>1.25)&&(C_UDMI(c1,t1,1)<2.75))
                                        {
                                                C_CENTROID(x1,c1,t1);
                                                NV_VV(x2,=,x,-,x1);
                                                NV_VS(x1,=,x2,/,NV_MAG(x2));

        /*flux=1025.*(rat)*(C_UDSI(c1,t1,1)+NV_DOT(C_UDSI_G(c1,t1,1),x2));*/
                                                if (NV_DOT(A,x1)>0)

        flux=1025.*(rat)*(C_UDSI(c1,t1,0)+NV_DOT(C_UDSI_G(c1,t1,0),x2))*NV_M
AG(A);
                                                else
                                                        flux=-
1025.*(rat)*(C_UDSI(c1,t1,0)+NV_DOT(C_UDSI_G(c1,t1,0),x2))*NV_MAG(A);

                                                C_UDMI(c1,t1,4)=1025.*(rat);
                                                flux=flux;
                                        }
                                }
                                else
                                {
                                        flux=F_FLUX(f,t);
                                }
                        }
                }
        }
        return flux;
}

DEFINE_EXECUTE_AT_END(exup2_sorce)
{
        Domain *d;
        Thread *t, *t0, *t1, *t2;
        face_t f1;
        cell_t c, c0, c1;
        real A[ND_ND],x[ND_ND],x1[ND_ND],x2[ND_ND],x3[ND_ND], cen, cen2,
con, conn, curv, col;
        real curvyn[596];
        real curvyt[596];
        real smth[600];
        int n, nl, loo, lo, loopl, cur;
        d = Get_Domain(1); /* mixture domain if multiphase */
        for (loopl=0;loopl<(TABLESIZE+1);loopl++)
        {
```

```
                curvyt[loopl]=0.;
                curvyn[loopl]=0.;
        }
        thread_loop_c(t,d)/*loop through cell
thread*//*&&(CURRENT_TIME>(718.*pow(C_STRAIN_RATE_MAG(c,t)+1.e-20,-
.2)))*/
        {
                if (FLUID_THREAD_P(t))
                {
                        begin_c_loop(c,t)
                        {

                                if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                                {
                                        con=0.;
                                        conn=0.;
                                        cen=0.;
                                        C_CENTROID(x1,c,t);
                                        c_face_loop(c, t, n)
                                        {
                                                f1=C_FACE(c,t,n);
                                                t1=C_FACE_THREAD(c,t,n);
                                                F_AREA(A,f1,t1);
                                                F_CENTROID(x2,f1,t1);
                                                if (x2[1]>cen)
                                                {
                                                        cen=x2[1];
                                                        if
(!BOUNDARY_FACE_THREAD_P(t1))
                                                        {
                                                                c1=F_C1(f1,t1);
                                                                t2=THREAD_T1(t1);
                                                                if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                                                                {
                                                                        NV_VV(x,=,x2,-,x1);

        NV_VS(x3,=,x,/,NV_MAG(x));

        con=C_UDMI(c,t,4)*(C_UDSI(c,t,0)+NV_DOT(C_UDSI_G(c,t,0),x))*NV_MA
G(A);
                                                                        if
(NV_DOT(C_UDSI_G(c,t,0),x3)<0)
                                                                        {

        conn=(NV_DOT(C_UDSI_G(c,t,0),x3))*NV_MAG(A);
```

```
                                                        }

                                                }
                                                c1=F_C0(f1,t1);
                                                t2=THREAD_T0(t1);
                                                if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                                                {
                                                        NV_VV(x,=,x2,-,x1);

        NV_VS(x3,=,x,/,NV_MAG(x));

        con=C_UDMI(c,t,4)*(C_UDSI(c,t,0)+NV_DOT(C_UDSI_G(c,t,0),x))*NV_MA
G(A);
                                                        if
(NV_DOT(C_UDSI_G(c,t,0),x3)<0)
                                                        {

        conn=(NV_DOT(C_UDSI_G(c,t,0),x3))*NV_MAG(A);
                                                        }

                                                }
                                        }
                                }
                                con=con/C_VOLUME(c,t)/1025.;
                                conn=-
C_UDSI_DIFF(c,t,0)*conn/C_VOLUME(c,t)/1025.;
                                if (con<conn)
                                        C_UDMI(c,t,3)=con;
                                else
                                        C_UDMI(c,t,3)=conn;
                                C_UDMI(c,t,3)=conn;
                                cur=((x1[0]/L+1.)*599/2.+.5);
                                smth[cur]=C_UDMI(c,t,3);

                        }
                        else
                        {
                                C_UDMI(c,t,3)=0.;
                        }
                }
                end_c_loop(c,t)
                        for (loo=0;loo<(10);loo++)
                        {
                                for (loopl=5;loopl<(595);loopl++)
```

```
                {
                        nl=0;
                        col=0.;
                        for (lo=-4;lo<5;lo++)
                        {
                                if (smth[loopl+lo]>0.)
                                {
                                        col=col+smth[loopl+lo];
                                        nl=nl+1;
                                }

                        }
                        smth[loopl]=col/(nl+1.e-20);
                }}
        begin_c_loop(c,t)
        {
                C_CENTROID(x,c,t);
                if (fabs(x[0])<L)
                {
                        cur=((x[0]/L+1.)*TABLESIZE/2.+.5);
                        /*C_UDMI(c,t,2)=curvy[cur];*/
                }
                /*if
((C_UDMI(c,t,2)>x[1])&&(C_UDMI(c,t,1)<.2))
                C_UDMI(c,t,5)=pow((ras-
C_UDMI(c,t,2))/ras,4)*C_VOLUME(c,t)*2.*M_PI/(5.e-14);
                else
                C_UDMI(c,t,5)=0.;     */
                if
(((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))&&(CURRENT_TIME>C_UDMI(c,t,1
0)))
                {
                        if (fabs(x[0])<L)
                        {
                                curvyt[cur]=curvyt[cur]+x[1];
                                curvyn[cur]=curvyn[cur]+1.;
                        }
                        cur=((x[0]/L+1.)*599/2.+.5);
                        if
((C_UDMI(c,t,0)>=0.2))/*&&(C_STRAIN_RATE_MAG(c,t)>1000.))
&&(CURRENT_TIME>(718.*pow(C_STRAIN_RATE_MAG(c,t)+1.e-20,-.2)))*/
                        {
                                C_UDMI(c,t,0)=C_UDMI(c,t,0)-
smth[cur]*CURRENT_TIMESTEP;
                                if (C_UDMI(c,t,0)<=.2)
                                {
```

```
                                                C_UDMI(c,t,0)=0.;
                                                C_UDMI(c,t,1)=3.;

        C_UDMI(c,t,6)=(C_VOLUME(c,t)*2.*M_PI);

        C_UDMI(c,t,8)=C_STRAIN_RATE_MAG(c,t);

        C_UDMI(c,t,9)=CURRENT_TIME;
                                                cen=0.;
                                                cen2=1.e20;
                                                c_face_loop(c, t, n)
                                                {
                                                        f1=C_FACE(c,t,n);

        t1=C_FACE_THREAD(c,t,n);

        F_CENTROID(x,f1,t1);
                                                        if (x[1]>cen)
                                                                cen=x[1];
                                                        if (x[1]<cen2)
                                                                cen2=x[1];
                                                        if ((100*(ras-
x[1])/ras)>stmin)

        stmin=(100*(ras-x[1])/ras);
                                                        if
(!BOUNDARY_FACE_THREAD_P(t1))
                                                        {

        c0=F_C1(f1,t1);

        t0=THREAD_T1(t1);
                                                                if
(C_UDMI(c0,t0,1)<.02)
                                                                {

        C_UDMI(c0,t0,1)=2;

        C_UDMI(c0,t0,10)=CURRENT_TIME;
                                                                }

        c0=F_C0(f1,t1);

        t0=THREAD_T0(t1);
                                                                if
(C_UDMI(c0,t0,1)<.02)
```

```
                                                                {

        C_UDMI(c0,t0,1)=2;

        C_UDMI(c0,t0,10)=CURRENT_TIME;
                                                                }
                                                            }
                                                        }
                                                        C_UDMI(c,t,7)=(cen-
cen2)/(CURRENT_TIME-C_UDMI(c,t,10)+.00001);
                                                    C_U(c,t)=0;
                                                    C_V(c,t)=0;
                                                }
                                            }
                                        }
                                        if ((C_UDMI(c,t,1)>2.5)&&(C_UDMI(c,t,1)<3.5))
                                        {
                                            con=0;
                                            c_face_loop(c, t, n)
                                            {
                                                f1=C_FACE(c,t,n);
                                                t1=C_FACE_THREAD(c,t,n);
                                                if (!BOUNDARY_FACE_THREAD_P(t1))
                                                {
                                                    c1=F_C1(f1,t1);
                                                    t2=THREAD_T1(t1);
                                                    if
((C_UDMI(c1,t2,1)>1.5)&&(C_UDMI(c1,t2,1)<2.5))
                                                    {
                                                        con=1;;
                                                    }
                                                    c1=F_C0(f1,t1);
                                                    t2=THREAD_T0(t1);
                                                    if
((C_UDMI(c1,t2,1)>1.5)&&(C_UDMI(c1,t2,1)<2.5))
                                                    {
                                                        con=1.;
                                                    }
                                                }
                                            }
                                            if (con<.5)
                                            {
                                                C_UDMI(c,t,1)=.45;
                                                C_UDSI(c,t,0)=0;
                                            }
                                        }
```

```
                        }
                        end_c_loop(c,t)
                }
        }
        for (loopl=0;loopl<(TABLESIZE+1);loopl++)
        {
                curv=(ras)-(ras)*st*.5*(1-cos(M_PI*(((L)*(2.*loopl/TABLESIZE-
1.))/(L)+1.)));

                if ((curvyt[loopl]>.0)&&(curvyt[loopl]/(curvyn[loopl]+.000001)<curv))
                {

                        curvy[loopl]=curvyt[loopl]/(curvyn[loopl]+.000001);
                }
                else
                        curvy[loopl]=curv;
                Message ("x: %e",curvy[loopl]);

        }
}
DEFINE_ON_DEMAND(calcup_flx)
{
        Domain *d;

Thread *t, *t0, *t1, *t2;
        face_t f1;
        cell_t c, c0, c1;
        real A[ND_ND],x[ND_ND],x1[ND_ND],x2[ND_ND],x3[ND_ND], cen, cen2,
con, conn, curv, col;
        real smth[600];
        real curvyn[596];
        real curvyt[596];
        int n, nl, loo, lo, loopl, cur;
        d = Get_Domain(1); /* mixture domain if multiphase */

        thread_loop_c(t,d)/*loop through cell
thread*//*&&(CURRENT_TIME>(718.*pow(C_STRAIN_RATE_MAG(c,t)+1.e-20,-
.2)))*/
        {
                if (FLUID_THREAD_P(t))
                {
                        begin_c_loop(c,t)
                        {
                                if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                                {
                                        con=0.;
```

```
                                conn=0;
                                cen=0.;
                                C_CENTROID(x1,c,t);
                                c_face_loop(c, t, n)
                                {
                                        f1=C_FACE(c,t,n);
                                        t1=C_FACE_THREAD(c,t,n);
                                        F_AREA(A,f1,t1);
                                        F_CENTROID(x2,f1,t1);
                                        if (x2[1]>cen)
                                        {
                                                cen=x2[1];
                                                if
(!BOUNDARY_FACE_THREAD_P(t1))
                                                {
                                                        c1=F_C1(f1,t1);
                                                        t2=THREAD_T1(t1);
                                                        if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                                                        {
                                                                NV_VV(x,=,x2,-,x1);

        NV_VS(x3,=,x,/,NV_MAG(x));

        con=C_UDMI(c,t,4)*(C_UDSI(c,t,0)+NV_DOT(C_UDSI_G(c,t,0),x))*NV_MA
G(A);
                                                                C_UDMI(c,t,5)=2;
                                                                if
(NV_DOT(C_UDSI_G(c,t,0),x3)<0)
                                                                {

        conn=(NV_DOT(C_UDSI_G(c,t,0),x3))*NV_MAG(A);

        /*C_UDMI(c,t,10)=((-
60.e6)/1025.)*C_UDSI_DIFF(c,t,1)*(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C
_UDSI(c,t,1)*NV_DOT(C_UDSI_G(c,t,0),x2));*/
                                                                }

        /*con=NV_MAG(A)*C_UDMI(c,t,4)*(C_UDSI(c,t,1));

        C_UDMI(c,t,10)=((60.e6)/1025.)*C_UDMI(c,t,4)*C_UDSI(c,t,1);
                                                                NV_VV(x2,=,x2,-
,x1);

        NV_VS(x2,=,x2,/,NV_MAG(x2));
```

```
                                                                    if
(NV_DOT(C_UDSI_G(c,t,1),x2)<0)
                                                                    {

        conn=(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C_UDSI(c,t,1)*NV_DO
T(C_UDSI_G(c,t,0),x2))*NV_MAG(A);
                                                                    C_UDMI(c,t,10)=((-
60.e6)/1025.)*C_UDSI_DIFF(c,t,1)*(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C
_UDSI(c,t,1)*NV_DOT(C_UDSI_G(c,t,0),x2));
                                                                    }*/
                                                                }
                                                                c1=F_C0(f1,t1);
                                                                t2=THREAD_T0(t1);
                                                                if
((C_UDMI(c1,t2,1)>2.5)&&(C_UDMI(c1,t2,1)<3.5))
                                                                {
                                                                    NV_VV(x,=,x2,-,x1);

        NV_VS(x3,=,x,/,NV_MAG(x));

        con=C_UDMI(c,t,4)*(C_UDSI(c,t,0)+NV_DOT(C_UDSI_G(c,t,0),x))*NV_MA
G(A);
                                                                    C_UDMI(c,t,5)=1;
                                                                    if
(NV_DOT(C_UDSI_G(c,t,0),x3)<0)
                                                                    {

        conn=(NV_DOT(C_UDSI_G(c,t,0),x3))*NV_MAG(A);

        /*C_UDMI(c,t,10)=((-
60.e6)/1025.)*C_UDSI_DIFF(c,t,1)*(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C
_UDSI(c,t,1)*NV_DOT(C_UDSI_G(c,t,0),x2));*/
                                                                    }

        /*con=NV_MAG(A)*C_UDMI(c,t,4)*.5*(C_UDSI(c,t,1));

        C_UDMI(c,t,10)=((60.e6)/1025.)*C_UDMI(c,t,4)*C_UDSI(c,t,1);
                                                                    NV_VV(x2,=,x2,-
,x1);

        NV_VS(x2,=,x2,/,NV_MAG(x2));
                                                                    if
(NV_DOT(C_UDSI_G(c,t,1),x2)<0)
                                                                    {
```

```
        conn=(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C_UDSI(c,t,1)*NV_DO
T(C_UDSI_G(c,t,0),x2))*NV_MAG(A);
                                                        C_UDMI(c,t,10)=((-
60.e6)/1025.)*C_UDSI_DIFF(c,t,1)*(C_UDSI(c,t,0)*NV_DOT(C_UDSI_G(c,t,1),x2)+C
_UDSI(c,t,1)*NV_DOT(C_UDSI_G(c,t,0),x2));
                                                        }*/
                                                }
                                        }
                                }
                        }
                        con=con/C_VOLUME(c,t)/1025.;
                        conn=-
C_UDSI_DIFF(c,t,0)*conn/C_VOLUME(c,t)/1025.;

        /*C_UDMI(c,t,3)=C_UDMI(c,t,5)/C_VOLUME(c,t)/1025.;*/
                        if (con<conn)
                                C_UDMI(c,t,3)=con;
                        else
                                C_UDMI(c,t,3)=conn;
                        cur=((x1[0]/L+1.)*599/2.+.5);
                        smth[cur]=C_UDMI(c,t,3);

                }
                else
                {
                        C_UDMI(c,t,3)=0.;
                }
        }
        end_c_loop(c,t)
                for (loo=0;loo<(10);loo++)
                {
                for (loopl=5;loopl<(595);loopl++)
                {
                        nl=0;
                        col=0.;
                        for (lo=-4;lo<5;lo++)
                        {
                                if (smth[loopl+lo]>0.)
                                {
                                        col=col+smth[loopl+lo];
                                        nl=nl+1;
                                }
                        }
                        smth[loopl]=col/(nl+1.e-20);
                }}
```

```
                begin_c_loop(c,t)
        {
                C_CENTROID(x,c,t);
                if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                {
                        cen=0.;
                        cen2=1.e20;
                        c_face_loop(c, t, n)
                        {
                                f1=C_FACE(c,t,n);
                                t1=C_FACE_THREAD(c,t,n);
                                F_CENTROID(x,f1,t1);
                                if (x[1]>cen)
                                        cen=x[1];
                                if (x[1]<cen2)
                                        cen2=x[1];
                        }
                        cur=((x[0]/L+1.)*599/2.+.5);
                        C_UDMI(c,t,2)=(60.e6)*(cen-cen2)*smth[cur]/.8;
                        C_UDMI(c,t,5)=smth[cur];
                }
                else
                        C_UDMI(c,t,2)=0.;
        }
        end_c_loop(c,t)
    }
}

}
DEFINE_ON_DEMAND(mem_8_init)
/*Used to initialize the user defined memory to specific values*/
{
        cell_t c;
        Domain *d;
        Domain *dom;
        Thread *t;
        face_t f, f1, f2, f3, fx, fy;
        Thread *t2, *t1, *t0, *tx, *t3, *ty, *tcy, *tcyy;
        cell_t c0, c1, cy, cyy;
        real x[ND_ND];
        real a[ND_ND];
        real curv, cen, cen2, cen3, area;
        int id, n, goo;
        int loopl;
        stmin=100.*st;
        for (loopl=0;loopl<(TABLESIZE+1);loopl++)
```

```
        {
                curvy[loopl]=(ras)-(ras)*st*.5*(1-
cos(M_PI*(((L)*(2.*loopl/TABLESIZE-1.))/(L)+1.)));
                Message ("x: %e",curvy[loopl]);
        }
        d=Get_Domain(1);
        thread_loop_c (t,d)
        {
                begin_c_loop (c,t)
                {
                        C_UDMI(c,t,0)=1.;
                        C_UDMI(c,t,1)=0.;
                        C_UDMI(c,t,2)=ras;
                        C_UDMI(c,t,3)=0.;
                        C_UDMI(c,t,4)=0.;
                        C_UDMI(c,t,5)=0.;
                        C_UDMI(c,t,6)=0.;
                        C_UDMI(c,t,7)=0.;
                        C_UDMI(c,t,8)=0.;
                        C_UDMI(c,t,9)=-1.;
                        C_UDMI(c,t,10)=0.;
                        C_CENTROID(x,c,t);
                        if (fabs(x[0])<=L)
                        {
                                int cur=((x[0]/L+1.)*TABLESIZE/2.+.5);
                                curv=(ras-ras*st*.5*(1.-cos(M_PI*(x[0]/L+1.))));
                                C_UDMI(c,t,2)=curvy[cur];
                                if (x[1]>curv)
                                {
                                        C_UDMI(c,t,1)=.5;
                                        C_UDMI(c,t,0)=0.;
                                }
                        }
                }
                end_c_loop (c,t)
        }
        thread_loop_c (t,d)
        {
                begin_c_loop (c,t)
                {
                        C_CENTROID(x,c,t);
                        if (fabs(x[0])<=(L))
                        {
                                if (C_UDMI(c,t,1)<.25)
                                {
                                        c_face_loop(c, t, n)
```

```
                                {
                                        f1=C_FACE(c,t,n);
                                        t1=C_FACE_THREAD(c,t,n);
                                        if (!BOUNDARY_FACE_THREAD_P(t1))
                                        {
                                                c1=F_C1(f1,t1);
                                                t2=THREAD_T1(t1);

if((C_UDMI(c1,t2,1)>.2)&&(C_UDMI(c1,t2,1)<.7))
                                                        C_UDMI(c,t,1)=2.;
                                                c1=F_C0(f1,t1);
                                                t2=THREAD_T0(t1);

if((C_UDMI(c1,t2,1)>.2)&&(C_UDMI(c1,t2,1)<.7))
                                                        C_UDMI(c,t,1)=2.;
                                        }
                                }
                        }
                }
        }
        end_c_loop (c,t)
}
thread_loop_c (t,d)
{
        begin_c_loop (c,t)
        {
                C_CENTROID(x,c,t);
                        if ((C_UDMI(c,t,1)>1.5)&&(C_UDMI(c,t,1)<2.5))
                        {
                                c_face_loop(c, t, n)
                                {
                                        f1=C_FACE(c,t,n);
                                        t1=C_FACE_THREAD(c,t,n);
                                        if (!BOUNDARY_FACE_THREAD_P(t1))
                                        {
                                                c1=F_C1(f1,t1);
                                                t2=THREAD_T1(t1);

if((C_UDMI(c1,t2,1)>.2)&&(C_UDMI(c1,t2,1)<.7))
                                                        C_UDMI(c1,t2,1)=3.;
                                                c1=F_C0(f1,t1);
                                                t2=THREAD_T0(t1);

if((C_UDMI(c1,t2,1)>.2)&&(C_UDMI(c1,t2,1)<.7))
                                                        C_UDMI(c1,t2,1)=3.;
                                        }
```

```
                                        }
                                }
                        }
                        end_c_loop (c,t)
                }
                /*t3 = Lookup_Thread(d,6);
                begin_f_loop(f3,t3)
                {
                        F_CENTROID(x,f3,t3);
                        if ((x[0]>(-5*ras))&&(x[0]<(20*ras)))
                        {
                                c0=F_C0(f3,t3);
                                t0=THREAD_T0(t3);
                                if ((C_UDMI(c0,t0,1)==0)&&(C_UDMI(c0,t0,0)>.02))
                                {
                                        C_UDMI(c0,t0,1)=2;
                                }
                        }
                }
                end_f_loop(f3,t3)*/
}
```

Appendix M: Creating Plots for Species Transport Model (Chapter 4)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Creating plots for species transport model%
%David Bark%
%11/11/10%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all
%
srr=[40 60 80];
tim1=[2800 1500 800]/60;
figure2=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
plot(srr,tim1,'Marker','o','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',10,'Lin
eStyle','none');
hold on
srr=[40:98];
tim1=.8*(1-srr/100)*.0015/((10^-4)*.004)/60;
plot(srr,tim1,'LineWidth',3,'LineStyle','-','Color',[.4 .4 .4]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.6    0.7250]);
grid on
xlabel('% Stenosis by Diameter','FontSize',35);
ylabel('Occlusion Time (minutes)','FontSize',35);
legend('Model','Eqn 19')%,'k=1x10^-^3','k=1x10^-^4','k=1x10^-^5'


shrr=[11.51 12.632  13.864  15.216  16.699  155.74  170.92  187.58  205.87  225.95
247.98  272.15  298.69  327.81  359.77  394.85  433.34  475.6   521.97  572.86  628.71
690.01  757.28  831.12  912.15  1001.1  1098.7  1205.8  1323.4  1452.4  1594    1749.4
1920    2107.2  2312.6  2538.1  2785.6  3057.2  3355.2  3682.4  4041.4  4435.4  4867.9
5342.5  5863.4  6435    7062.4  7751    8506.7  9336.1  10246   11245   12342   13545
14866   16315   17906   19652   21568   23670   25978   28511   31291   34342   37690
41365   45398   49824   54682   60013   65865   72286   79334   87069   1.0978e+005];
grwr=(8.8/.8)*[-0.12291 -0.0016241  0.019925    0.14017 0.15829 0.44832 0.31085
0.40613 0.34395 0.3324  0.35499 0.48932 0.40343 0.46807 0.61332 0.64955 0.54485
0.62165 0.64919 0.61853 0.53858 0.67235 0.687   0.96153 0.87828 1.004   0.97622
0.86948 0.88824 1.0598  1.2514  1.3531  1.2256  1.2485  1.1756  1.2876  1.4534  1.6977
1.5796  1.5046  1.5545  1.7287  1.7088  1.9616  1.9272  1.8692  1.8292  1.7678  1.6853
1.624   1.5629  1.6183  1.5752  1.5087  1.6152  1.615   1.6206  1.4734  1.5456  1.36
1.3558  1.4431  1.252   1.0039  1.0022  0.91955 1.1194  1.1812  0.94486 1.1516  1.0386
1.4 1.4194  1.551   1.9953];
sr=[11.51   12.632  13.864  15.216  16.699  18.327  20.114  22.075  24.227  26.59
29.182  32.027  35.15   38.577  42.338  46.466  50.997  55.969  61.425  67.414  73.987
81.201  89.118  97.807  107.34  117.81  129.29  141.9   155.74  170.92  187.58  205.87
```

```
225.95  247.98  272.15  298.69  327.81  359.77  394.85  433.34  475.6   521.97  572.86
628.71  690.01  757.28  831.12  912.15  1001.1  1098.7  1205.8  1323.4  1452.4  1594
1749.4  1920    2107.2  2312.6  2538.1  2785.6  3057.2  3355.2  3682.4  4041.4  4435.4
4867.9  5342.5  5863.4  6435    7062.4  7751    8506.7  9336.1  10246   11245   12342
13545   14866   16315   17906   19652   21568   23670   25978   28511   31291   34342
37690   41365   45398   49824   54682   60013   65865   72286   79334   87069
1.0978e+005];
gr=[0.85393 -0.14311    0.15109 0.36021 0.52763 0.99521 1.3383  1.616   1.9966
0.77251 1.4092  1.1444  4.1272  5.7017  2.9504  1.522   7.1506  5.9721  5.8123  4.7114
5.4881  3.6612  3.6719  4.0879  7.4856  5.7629  4.7944  4.6575  4.8419  7.7598  12.505
9.0797  3.9051  4.0031  5.7086  5.7916  4.6083  6.0816  7.6048  5.6263  6.5464  7.615
7.1998  6.7646  9.024   8.6558  10.599  9.7119  10.943  10.37   9.9651  10.797  11.749
13.501  14.223  13.82   15.028  13.999  14.647  15.504  17.066  15.59   15.584  16.782
17.85   18.598  20.873  21.142  20.477  19.772  18.833  19.059  18.012  17.563  18.197
18.062  17.748  18.182  18.154  17.381  16.749  17.018  15.408  15.344  14.82   14.147
12.006  10.681  10.222  11.012  11.693  9.6662  10.233  10.907  15.4    15.614  17.061
21.948];
grb=[6.4182 2.4437  1.8018  1.8581  1.7119  1.5339  3.8315  1.965   7.0162  4.0763
3.395   4.0645  3.2574  12.462  9.6647  3.3957  17.433  16.41   15.33   11.441  12.636
10.447  9.2061  8.0942  18.146  13.659  12.174  13.777  12.645  13.356  23.048  18.881
8.4407  7.5351  10.695  10.933  7.103   8.7901  13.201  8.2141  8.9223  10.633  9.454
9.8163  12.211  10.882  9.1193  10.399  10.872  9.3862  10.767  11.782  11.032  11.35
11.297  11.348  10.382  9.9854  10.603  9.9828  9.7374  8.1953  8.1149  9.2457  8.3193
9.574   8.91    8.9995  9.3518  10.185  9.2563  9.5989  9.478   9.6416  9.9896  10.098
8.9384  9.4181  9.2516  8.2726  8.0392  8.6831  8.3192  8.3212  8.1903  7.7523  7.482
7.4446  7.8623  8.9005  8.7192  7.0884  7.8998  7.0839  9.2034  12.108  9.8551  11.615];
 shrbin=logspace(1,5,60);
sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
% figure2=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
% % semilogx(shrr,grwr,'LineWidth',5,'LineStyle','-','Color',[.6 .6
.6]);%micrometer/minute
% hold on
% %
% y_nam='C:\Documents and Settings\Dave\My Documents\ppr_3_figs\e5';
% % k=[1 2 5 6 7 100 3 4];15 3 4 209 409 4090 40900 11 111 12
% symb=['o','s','^','>','<','v','.','*','+'];
% k=0;
% ti=[1000 1000 1000 1000 1000];
% for j=[120];
%
% %     if (j<1000)
% %         k=k+1;
% %     end
% k=k+1;
%     fid = fopen([y_nam,num2str(j)],'r');
%         shrdat = textscan(fid,'%f %f %f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
```

```
%     shrdat=cell2mat(shrdat);
%     xl=(shrdat(:,2));
%     yl=(shrdat(:,3));
%     occl_marker=(shrdat(:,4));
%     g_rate=abs(shrdat(:,5));
%     strain=(shrdat(:,6));
% %     tim=(shrdat(:,8));
%     fclose(fid);
%
%     occl_mark=[1.5 2.5];
%     [nn,xbins]=histc(occl_marker,occl_mark);%xbins is the bin location
%     str=strain(xbins==1);
%     grw=g_rate(xbins==1);
%     xl=xl(xbins==1);
%     yl=yl(xbins==1);
%     occl_marker=occl_marker(xbins==1);
% %     tim=tim(xbins==1);
%         occl_mark=[-1000 0];
% %     [nn,xbins]=histc(xl,occl_mark);%xbins is the bin location
% %     str=strain(xbins==1);
% %     grw=g_rate(xbins==1);
% %     xl=xl(xbins==1);
% %     yl=yl(xbins==1);
% %     occl_marker=occl_marker(xbins==1);
%
%     [nn,xbins]=histc(str,shrbin);%xbins is the bin location
%
%     gbin=[];
% %     gbinn=[];
%     sb2=[];
%     for i=[1:length(sb)]
%         if (sum(grw(xbins==i))~=0)
%             gbin=[gbin (mean(grw(xbins==i)))];%averaged growth
%             %         gbinn=[gbinn std((vol(xbins==i)))];%./sum(xbins==i)];%averaged
growth
%             sb2=[sb2 sb(i)];
%         end
%     end
%
plot(xl/max(yl),grw,'Marker',symb(k),'MarkerEdgeColor','k','MarkerFaceColor','none','M
arkerSize',10,'LineStyle','none');
% end
%
% [x ix]=sort(xl);
% y=grw(ix);
% yy=y;
```

```
% for i=1:10
%     yy=smooth(yy,'moving');
% end
% figure
% plot(x,y,'.')
% hold on
% plot(x,yy)
%
%
% yy=y;
% yyy=yy;
% for i=1:10
%     for j=5:(length(yy)-5)
%         yyy(j)=mean(yy([(j-4):(j+4)]));
%     end
%     yy=yyy;
% end
% figure
% plot(x,y,'.')
% hold on
% plot(x,yy,'r')
%
% % xl=xl-min(xl);
% % % semilogx(str,grw,'s')
% % J=(60e6)*((.002/1.8575)*((((7e-13)*shrr+1.6e-13).^2).*shrr./(2.5e-3)).^(1/3))./.8;
% % kt=2e-4;
% % J=1.8575./((((7e-13).*shrr+1.6e-13).^2).*shrr./(2.5e-3)).^(1./3);
% % J=J+1./kt;
% % J=(60e6).*.002./J;
% % % J=(60e6).*((.002).*(1./kt+1.8575./((shrr.*(((7e-13).*shrr+1.6e-13).^2)./(2.5e-
3)).^(1./3))))
% % % J=(60e6)*(.002/1.8575)*((((7e-13)*shrbin+1.6e-13).^2).*shrbin/.01).^(1/3)/.8;
% % semilogx(shrr,J,'LineWidth',5,'LineStyle','--','Color',[.4 .4
.4]);%micrometer/minute%

etim=[0 10  20  30  40  50  60  70  80  90  100 110 120 130 140 150 160 170 180 190 200
210 220 230 240 250 260];
evol=[0 0.34786 -    0.11973i   0.77775 -   0.034984i   1.1705 -    0.31347i    1.5608 -
0.62743i    2.0396 -    0.94694i    2.5636 -     1.3971i    3.4198 -     1.7184i    4.7839 -
2.1816i    6.2666 -     2.6915i    8.5026 -     2.9961i    11.009 -     3.4003i    13.871 -
3.9367i    16.95 -     4.0542i 20.18 -     4.2903i 23.816 -     4.6025i    27.298 -     4.7942i
30.36 -     4.9881i 33.665 -     5.0231i    36.835 -     4.9744i    40.625 -     5.1519i    44.398
-     5.2556i    47.043 -     5.5786i    50.502 -     5.7532i    53.282 -      5.877i    56.387 -
6.0823i    59.89 -     6.1408i];
```

```
estd=sqrt(6)*[0 0.22269 0.36116 0.40437 0.52602 0.71367 0.59514 0.59886 0.83802
1.1648  1.8357  2.498   2.8808  3.0321  3.3064  3.229   3.0957  2.8712  2.4791  2.0605
2.4803  2.4325  2.534   3.0134  3.7695  4.0703  4.9977];
% estd=[0   0.72634 1.0799  1.2522  1.7644  2.1918  2.0538  1.9834  3.2289  4.0003
5.2752  6.7282  7.7502  8.066   8.6974  8.6236  8.4336  7.9406  7.0693  6.3172  7.1533
6.3314  6.9833  7.6318  9.2749  10.308  12.026  14.266  15.575  17.314  39.812  42.006
54.257  55.783  60.05   63.702  66.727  72.103  74.695  46.323  44.644  41.626];
figure2=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
ax1=plot(etim,evol,'LineWidth',3,'LineStyle','-','Color',[.4 .4 .4]);
hold on
% ax2=plot(etim,evol+estd,'LineWidth',2,'LineStyle','--','Color',[.7 .7 .7]);

timm=[0:10:300];
timmm=[5:10:295];

y_nam='C:\Documents and Settings\Dave\My Documents\ppr_3_figs\e';
% k=[1 2 5 6 7 100 3 4];15 3 4 10 3 4 5 24
symb=['o','s','^','>','<','v','.','*','+'];
k=0;
ti=[1000 1000 1000 1000 1000];
for j=[54 541];
   k=k+1;
   fid = fopen([y_nam,num2str(j)],'r');
      shrdat = textscan(fid,'%f %f %f %f %f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
   shrdat=cell2mat(shrdat);
   xl=(shrdat(:,2));
   yl=(shrdat(:,3));
   occl_marker=(shrdat(:,4));
   vol=abs(shrdat(:,5));
   g_rate=(60e6)*abs(shrdat(:,6));
   strain=(shrdat(:,7));
   tim=(shrdat(:,8));
   fclose(fid);

   occl_mark=[.42 .47];
   [nn,xbins]=histc(occl_marker,occl_mark);%xbins is the bin location
   str=strain(xbins==1);
   grw=g_rate(xbins==1);
   vol=vol(xbins==1);
   xl=xl(xbins==1);
   yl=yl(xbins==1);
   occl_marker=occl_marker(xbins==1);
   tim=tim(xbins==1);

   occl_mark=[-1 ti(k)];
   [nn,xbins]=histc(tim,occl_mark);%xbins is the bin location
```

```
    str=str(xbins==1);
    grw=grw(xbins==1);
    vol=vol(xbins==1);
    xl=xl(xbins==1);
    yl=yl(xbins==1);
    occl_marker=occl_marker(xbins==1);
    tim=tim(xbins==1);

    [nn,xbins]=histc(tim,timm);%xbins is the bin location


gbin=[0];
gbinn=[0];
sb2=[0];
for i=[1:length(timmm)]
    gbin=[gbin (sum(vol(xbins==i))+gbin(i))];%averaged growth
    sb2=[sb2 timmm(i)];
end

plot(sb2,gbin*(1e12),'Marker',symb(k),'MarkerEdgeColor','k','MarkerFaceColor','k','Mark
erSize',10,'LineStyle','none');
end

% ax9=plot(etim,evol-estd,'LineWidth',2,'LineStyle','--','Color',[.7 .7 .7]);
hold off
xlim([0 250]);
set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.6    0.7250]);
grid on
xlabel('Time (s)','FontSize',35);
ylabel('Thrombus Growth (million microns^3)','FontSize',35);
legend('Exp Vol','k=1x10^-^4' ,'k=1x10^-^4 w/ lag','Location','NorthWest')%,'k=1x10^-
^3','k=1x10^-^4','k=1x10^-^5'

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
shrbin=logspace(1,5,60);
sb=mean([shrbin([2:(length(shrbin)-2)]);shrbin([3:(length(shrbin)-1)])]);%x bin values
egrw=(double(shrbin<=6000).*.41.*(shrbin.^.45)+double(shrbin>6000).*(20-(2.1e-
4)*shrbin));


shrr=[205.87    225.95  247.98  272.15  298.69  327.81  359.77  394.85  433.34  475.6
521.97  572.86  628.71  690.01  757.28  831.12  912.15  1001.1  1098.7  1205.8  1323.4
1452.4  1594    1749.4  1920    2107.2  2312.6  2538.1  2785.6  3057.2  3355.2  3682.4
```

```
4041.4  4435.4  4867.9  5342.5  5863.4  6435    7062.4  7751    8506.7  9336.1  10246
11245   12342   13545   14866   16315   17906   19652   21568   23670   25978   28511
31291   34342   37690   41365   45398   49824   54682   60013   65865   72286   79334
87069   1.0978e+005];
grwr=(8.8/.8)*[0.40216  0.37506 0.56361 0.59281 0.50177 0.59618 0.57855 0.65321
0.65891 0.6407  0.71822 0.61986 0.46063 0.57643 0.59638 0.90414 0.82081 0.99425
0.95874 0.84399 0.84941 1.037   1.2275  1.3557  1.1991  1.2382  1.1681  1.2733  1.4247
1.7453  1.6801  1.6449  1.6703  1.8184  1.7463  1.9734  1.9321  1.8692  1.8292  1.7678
1.6853  1.624   1.5629  1.6183  1.5752  1.5087  1.6152  1.615   1.6206  1.4734  1.5456
1.36    1.3558  1.4431  1.252   1.0039  1.0022  0.91955 1.1194  1.1812  0.94486 1.1516
1.0386  1.4 1.4194  1.551   1.9953];
shrr=[11.51 12.632  13.864  15.216  16.699  155.74  170.92  187.58  205.87  225.95
247.98  272.15  298.69  327.81  359.77  394.85  433.34  475.6   521.97  572.86  628.71
690.01  757.28  831.12  912.15  1001.1  1098.7  1205.8  1323.4  1452.4  1594    1749.4
1920    2107.2  2312.6  2538.1  2785.6  3057.2  3355.2  3682.4  4041.4  4435.4  4867.9
5342.5  5863.4  6435    7062.4  7751    8506.7  9336.1  10246   11245   12342   13545
14866   16315   17906   19652   21568   23670   25978   28511   31291   34342   37690
41365   45398   49824   54682   60013   65865   72286   79334   87069   1.0978e+005];
grwr=(8.8/.8)*[-0.12291 -0.0016241  0.019925    0.14017 0.15829 0.44832 0.31085
0.40613 0.34395 0.3324  0.35499 0.48932 0.40343 0.46807 0.61332 0.64955 0.54485
0.62165 0.64919 0.61853 0.53858 0.67235 0.687   0.96153 0.87828 1.004   0.97622
0.86948 0.88824 1.0598  1.2514  1.3531  1.2256  1.2485  1.1756  1.2876  1.4534  1.6977
1.5796  1.5046  1.5545  1.7287  1.7088  1.9616  1.9272  1.8692  1.8292  1.7678  1.6853
1.624   1.5629  1.6183  1.5752  1.5087  1.6152  1.615   1.6206  1.4734  1.5456  1.36
1.3558  1.4431  1.252   1.0039  1.0022  0.91955 1.1194  1.1812  0.94486 1.1516  1.0386
1.4 1.4194  1.551   1.9953];
figure2=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
semilogx(sr,gr,'LineWidth',3,'LineStyle','-','Color',[.4 .4 .4]);%micrometer/minute
hold on
% semilogx(sr,gr+grb,'LineWidth',2,'LineStyle','--','Color',[.7 .7 .7]);%micrometer/minute
%
y_nam='C:\Documents and Settings\Dave\My Documents\ppr_3_figs\e';
% k=[1 2 5 6 7 100 3 4];15 3 4 3 4 5 24 101 105 106 3 4 5 24 101 105 106
symb=['o','s','^','>','<','v','.','*','+'];
k=0;
ti=[100000 100000 100000 1000 1000 1000 1000 1000];
for j=[53 54 55];
   k=k+1;
   fid = fopen([y_nam,num2str(j)],'r');
      shrdat = textscan(fid,'%f %f %f %f %f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
   shrdat=cell2mat(shrdat);
   xl=(shrdat(:,2));
   yl=(shrdat(:,3));
   occl_marker=(shrdat(:,4));
   g_rate=(60e6)*abs(shrdat(:,6));
   strain=(shrdat(:,7));
```

```
    tim=(shrdat(:,8));
    fclose(fid);

    occl_mark=[.42 .47];
    [nn,xbins]=histc(occl_marker,occl_mark);%xbins is the bin location
    str=strain(xbins==1);
    grw=g_rate(xbins==1);
    xl=xl(xbins==1);
    yl=yl(xbins==1);
    occl_marker=occl_marker(xbins==1);
    tim=tim(xbins==1);

    occl_mark=[-1 ti(k)];
    [nn,xbins]=histc(tim,occl_mark);%xbins is the bin location
    str=str(xbins==1);
    grw=grw(xbins==1);
    xl=xl(xbins==1);
    yl=yl(xbins==1);
    occl_marker=occl_marker(xbins==1);
    tim=tim(xbins==1);

    [nn,xbins]=histc(str,shrbin);%xbins is the bin location

    gbin=[];
%     gbinn=[];
    sb2=[];
    for i=[1:length(sb)]
        if (sum(grw(xbins==i))~=0)
            gbin=[gbin (mean(grw(xbins==i)))];%averaged growth
            %         gbinn=[gbinn std((vol(xbins==i)))];%./sum(xbins==i)];%averaged growth
            sb2=[sb2 sb(i)];
        end
    end


semilogx(sb2,gbin,'Marker',symb(k),'MarkerEdgeColor','k','MarkerFaceColor','k','Marker
Size',10,'LineStyle','none');
end
% semilogx(sr,gr-grb,'LineWidth',2,'LineStyle','--','Color',[.7 .7 .7]);%micrometer/minute

xl=xl-min(xl);
% semilogx(str,grw,'s')
J=(60e6)*((.002/1.8575)*((((7e-13)*shrr+1.6e-13).^2).*shrr./(2.5e-3)).^(1/3))./.8;
kt=10^-4;
J=1.8575./((((7e-13).*shrr+1.6e-13).^2).*shrr./(2.5e-3)).^(1./3);
J=J+1./kt;
```

```
J=2*(60e6).*.002./J;
% J=(60e6).*((.002).*(1./kt+1.8575./((shrr.*(((7e-13).*shrr+1.6e-13).^2)./(2.5e-
3)).^(1./3))))
% J=(60e6)*(.002/1.8575)*((((7e-13)*shrbin+1.6e-13).^2).*shrbin/.01).^(1/3)/.8;
semilogx(shrr,J,'LineWidth',5,'LineStyle','--','Color',[.4 .4 .4]);%micrometer/minute


ylim([10^-2 10^2])
ylim([0 30])
xlim([100 100000])

set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.6    0.7250]);
grid on
xlabel('Shear rate (1/s)','FontSize',35);
ylabel({'Thrombus Growth Rate','(\mum^3/\mum^2-min)'},'FontSize',35);
legend('Exp Rate','k=1x10^-^3','k=1x10^-^4','k=1x10^-
^5','Analytical','Location','NorthWest')%,'k=1x10^-^4','k=1x10^-^5'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
figure2=figure('units','normalized','position',[0.05 .1 .9 .9],'Color',[1 1 1]);
semilogx(sr,gr,'LineWidth',5,'LineStyle','-','Color',[.6 .6 .6]);%micrometer/minute
hold on
% semilogx(sr,gr+grb,'LineWidth',2,'LineStyle','--','Color',[.7 .7 .7]);%micrometer/minute
%
y_nam='C:\Documents and Settings\Dave\My Documents\ppr_3_figs\e5';
% k=[1 2 5 6 7 100 3 4];15 3 4  1
symb=['o','s','^','>','<','v','.','*','+'];-2
k=0;
ti=[1000 1000 1000];
for j=[0 1 3 7];
   k=k+1;
   fid = fopen([y_nam,num2str(j)],'r');
      shrdat = textscan(fid,'%f %f %f %f %f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
   shrdat=cell2mat(shrdat);
   xl=(shrdat(:,2));
   yl=(shrdat(:,3));
   occl_marker=(shrdat(:,4));
   g_rate=(60e6)*abs(shrdat(:,6));
   strain=(shrdat(:,7));
%     tim=(shrdat(:,8));
   fclose(fid);

   occl_mark=[.42 .47 2.5 3.5];
   [nn,xbins]=histc(occl_marker,occl_mark);%xbins is the bin location
```

```
   str=strain((xbins==1)|(xbins==3));
   grw=g_rate((xbins==1)|(xbins==3));
   xl=xl((xbins==1)|(xbins==3));
   yl=yl((xbins==1)|(xbins==3));
   occl_marker=occl_marker((xbins==1)|(xbins==3));
%    tim=tim(xbins==1);

%    occl_mark=[-1 ti(k)];
%    [nn,xbins]=histc(tim,occl_mark);%xbins is the bin location
%    str=str(xbins==1);
%    grw=grw(xbins==1);
%    xl=xl(xbins==1);
%    yl=yl(xbins==1);
%    occl_marker=occl_marker(xbins==1);
%    tim=tim(xbins==1);

   [nn,xbins]=histc(str,shrbin);%xbins is the bin location

   gbin=[];
%    gbinn=[];
   sb2=[];
   for i=[1:length(sb)]
      if (sum(grw(xbins==i))~=0)
         gbin=[gbin (mean(grw(xbins==i)))];%averaged growth
         %        gbinn=[gbinn std((vol(xbins==i)))];%./sum(xbins==i)];%averaged growth
         sb2=[sb2 sb(i)];
      end
   end


semilogx(sb2,gbin,'Marker',symb(k),'MarkerEdgeColor','k','MarkerFaceColor','k','Marker
Size',10,'LineStyle','none');
end
% y_nam='C:\Documents and Settings\Dave\My Documents\ppr_3_figs\e3e';
% ti=[1000];
% for j=[0];
%    k=k+1;
%    fid = fopen([y_nam,num2str(j)],'r');
%       shrdat = textscan(fid,'%f %f %f %f %f %f %f ',-1, 'delimiter' , ',','headerlines', 1);
%    shrdat=cell2mat(shrdat);
%    xl=(shrdat(:,2));
%    yl=(shrdat(:,3));
%    occl_marker=(shrdat(:,4));
%    g_rate=(60e6)*abs(shrdat(:,6));
%    strain=(shrdat(:,7));
% %    tim=(shrdat(:,8));
```

```
%     fclose(fid);
%
%     occl_mark=[.42 .47];
%     [nn,xbins]=histc(occl_marker,occl_mark);%xbins is the bin location
%     str=strain(xbins==1);
%     grw=g_rate(xbins==1);
%     xl=xl(xbins==1);
%     yl=yl(xbins==1);
%     occl_marker=occl_marker(xbins==1);
% %     tim=tim(xbins==1);
%
% %     occl_mark=[-1 ti(k)];
% %     [nn,xbins]=histc(tim,occl_mark);%xbins is the bin location
% %     str=str(xbins==1);
% %     grw=grw(xbins==1);
% %     xl=xl(xbins==1);
% %     yl=yl(xbins==1);
% %     occl_marker=occl_marker(xbins==1);
% %     tim=tim(xbins==1);
%
%     [nn,xbins]=histc(str,shrbin);%xbins is the bin location
%
%     gbin=[];
% %     gbinn=[];
%     sb2=[];
%     for i=[1:length(sb)]
%         if (sum(grw(xbins==i))~=0)
%             gbin=[gbin (mean(grw(xbins==i)))];%averaged growth
%             %         gbinn=[gbinn std((vol(xbins==i)))];%./sum(xbins==i)];%averaged
growth
%             sb2=[sb2 sb(i)];
%         end
%     end
%
%
semilogx(sb2,gbin,'Marker',symb(k),'MarkerEdgeColor','k','MarkerFaceColor','none','Mar
kerSize',10,'LineStyle','none');
% end
% y_nam='C:\Documents and Settings\Dave\My Documents\ppr_3_figs\e5';
% ti=[1000];
% for j=[51];
%     k=k+1;
%     fid = fopen([y_nam,num2str(j)],'r');
%         shrdat = textscan(fid,'%f %f %f %f %f %f',-1, 'delimiter' , ',','headerlines', 1);
%     shrdat=cell2mat(shrdat);
%     xl=(shrdat(:,2));
```

```
%     yl=(shrdat(:,3));
%     occl_marker=(shrdat(:,4));
%     g_rate=abs(shrdat(:,5));
%     strain=(shrdat(:,6));
% %     tim=(shrdat(:,8));
%     fclose(fid);
%
%     occl_mark=[1.5 2.5];
%     [nn,xbins]=histc(occl_marker,occl_mark);%xbins is the bin location
%     str=strain(xbins==1);
%     grw=g_rate(xbins==1);
%     xl=xl(xbins==1);
%     yl=yl(xbins==1);
%     occl_marker=occl_marker(xbins==1);
% %     tim=tim(xbins==1);
%
%
%     [nn,xbins]=histc(str,shrbin);%xbins is the bin location
%
%     gbin=[];
% %     gbinn=[];
%     sb2=[];
%     for i=[1:length(sb)]
%         if (sum(grw(xbins==i))~=0)
%             gbin=[gbin (mean(grw(xbins==i)))];%averaged growth
%             %         gbinn=[gbinn std((vol(xbins==i)))];%./sum(xbins==i)];%averaged
growth
%             sb2=[sb2 sb(i)];
%         end
%     end
%
%
semilogx(sb2,gbin,'Marker',symb(k),'MarkerEdgeColor','k','MarkerFaceColor','none','Mar
kerSize',10,'LineStyle','none');
% end

xl=xl-min(xl);
% semilogx(str,grw,'s')
J=(60e6)*((.002/1.8575)*((((7e-13)*shrr+1.6e-13).^2).*shrr./(2.5e-3)).^(1/3))./.8;
kt=2e-1;
J=1.8575./((((7e-13).*shrr*0.+1.6e-13).^2).*shrr./(2.5e-3)).^(1./3);
J=J+1./kt;
J=2*(60e6).*.002./J;
% J=(60e6).*((.002).*(1./kt+1.8575./((shrr.*(((7e-13).*shrr+1.6e-13).^2)./(2.5e-
3)).^(1./3))))
% J=(60e6)*(.002/1.8575)*((((7e-13)*shrbin+1.6e-13).^2).*shrbin/.01).^(1/3)/.8;
```

```
semilogx(shrr,J/.8,'LineWidth',5,'LineStyle','--','Color',[.4 .4 .4]);%micrometer/minute

%
% kt=(2e-4)-(1e-9).*shrr;
% J=1.8575./((((7e-13).*shrr+1.6e-13).^2).*shrr./(2.5e-3)).^(1./3);
% J=J+1./kt;
% J=(60e6).*.002./J;
% % J=(60e6).*((.002).*(1./kt+1.8575./((shrr.*(((7e-13).*shrr+1.6e-13).^2)./(2.5e-
3)).^(1./3))))
% % J=(60e6)*(.002/1.8575)*((((7e-13)*shrbin+1.6e-13).^2).*shrbin/.01).^(1/3)/.8;
% semilogx(shrr,J,'LineWidth',5,'LineStyle','-.','Color',[.4 .4 .4]);%micrometer/minute
% semilogx(sr,gr-grb,'LineWidth',2,'LineStyle','-.','Color',[.7 .7 .7]);%micrometer/minute
ylim([10^-2 10^2])
ylim([0 30])
xlim([100 10000])

set(gca,'FontSize',30);
set(gca,'position',[0.1893    0.2000    0.6    0.7250]);
grid on
xlabel('Shear rate (1/s)','FontSize',35);
ylabel({'Thrombus Growth Rate','(\mum^3/\mum^2-min)'},'FontSize',35);
legend('Exp Rate','Uniform RBC','Thermal Diffusivity','k=1x10^-
^3','Analytical','Location','NorthWest')
```