
Theses and Dissertations

Spring 2016

Building user interactive capabilities for image-based modeling of patient-specific biological flows in single platform

Liza Shrestha
University of Iowa

Copyright 2016 Liza Shrestha

This dissertation is available at Iowa Research Online: <http://ir.uiowa.edu/etd/3189>

Recommended Citation

Shrestha, Liza. "Building user interactive capabilities for image-based modeling of patient-specific biological flows in single platform." PhD (Doctor of Philosophy) thesis, University of Iowa, 2016. <http://ir.uiowa.edu/etd/3189>.

Follow this and additional works at: <http://ir.uiowa.edu/etd>



Part of the [Biomedical Engineering and Bioengineering Commons](#)

BUILDING USER INTERACTIVE CAPABILITIES FOR IMAGE-BASED-
MODELING OF PATIENT-SPECIFIC BIOLOGICAL FLOWS IN SINGLE
PLATFORM

by

Liza Shrestha

A thesis submitted in partial fulfillment
of the requirements for the Doctor of Philosophy
degree in Biomedical Engineering in the
Graduate College of
The University of Iowa

May 2016

Thesis Supervisor: Associate Professor Sarah C. Vigmostad

Copyright by
LIZA SHRESTHA
2016
All Rights Reserved

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Liza Shrestha

has been approved by the Examining Committee for
the thesis requirement for the Doctor of Philosophy degree
in Biomedical Engineering at the May 2016 graduation.

Thesis Committee:

Sarah C. Vigmostad, Thesis Supervisor

H. S. Udaykumar

Nicole Grosland

Vincent A. Magnotta

M. L. Raghavan

To the forces that made this journey possible.

ACKNOWLEDGEMENTS

I would like to thank my family and friends who believed in me. I would like to thank my advisor who gave me the opportunity. I would like to thank my committee members who guided me. Finally, I would like to thank my angels who were always with me.

ABSTRACT

In this work, we have developed user interactive capabilities that allow us to perform segmentation and manipulation of patient-specific geometries required for Computational Fluid Dynamics (CFD) studies, entirely in image domain and within a single platform of 'IAFEMesh'. Within this toolkit we have added commonly required manipulation capabilities for performing CFD on segmented objects by utilizing libraries like ITK, VTK and KWWidgets. With the advent of these capabilities we can now manipulate a single patient specific image into a set of possible cases we seek to study; which is difficult in commercially available software like VMTK, Slicer, MITK etc. due to their limited manipulation capabilities. Levelset representation of the manipulated geometries can be simulated in our flow solver without creating any surface or volumetric mesh. This image-levelset-flow framework offers few advantages. 1) We don't need to deal with the problems associated with mesh quality, edge connectivity related to mesh models, 2) and manipulations operations result in smooth, physically realizable entities which is challenging in mesh domain. We have validated our image-levelset-flow setup by simulating and validating the results of flow in idealized geometries like cylinder with empirical results and patient-specific flow in Intracranial aneurysm (ICA) with the known results from previous studies. We further demonstrated the utility of the toolkit by studying the effect of angle of flow extensions, and effect of aneurysm in Middle Cerebral Artery (MCA).

The capabilities developed in this work enabled us to perform CFD simulations to understand the hemodynamics in Type-A aortic dissection. An aortic dissection is a life threatening disease that occurs due to the tearing of intimal layer of aorta resulting in the

formation of false lumen which provides an alternate channel for blood to flow. If the dissection exists in ascending aorta, then it is classified as Type-A aortic dissection. We implemented a new segmentation algorithm by following closely the work of Krissian et al. (2014) and modifying it to suit our needs. We used CT images of a patient with Type-A dissection who had a primary entry tear, a secondary tear and tertiary exit tears. We developed permutations of the diseased aorta to study the effect of Type-A dissection in hemodynamics in aorta. We performed pulsatile flow simulations using flow data using patient specific flow data from another patient. We compared the hemodynamics in the dissected aorta with that in a baseline ‘healthy’ aorta, created by virtually removing the dissection and merging the true and false lumens. We observed increased velocity, pressure and wall shear stress in the narrowed lumens and around the primary tear. We investigated the impact of repair surgery by preparing a model of Type-A dissection by occluding the primary entry tear and measuring the flow in true and false lumen. We found that repair surgery significantly reduced the retrograde flow in the false lumen. The reduction was about 99% in arch and cross sections closer to arch in the descending aorta. We created four instances of Type-A dissection by altering the location of the primary entry tear and studied the effect of location of the primary tear. The highest pressure and velocity were observed when the primary entry tear was closest to the ascending aorta. Finally, we also studied the impact of the secondary tear, which revealed that hemodynamics in the flow distal to the secondary tear was clearly compromised when the tear was not modeled but it did not have significant impact in the vicinity of the arch.

In summary, we have successfully built an image based framework for segmentation and manipulation of patient-specific images. We have modified the

algorithm by Krissian et al. and implemented it for the segmentation of Type-A aortic dissection. Finally, we implemented these capabilities to study the hemodynamics in Type-A aortic dissection. Our image based framework is a first of its kind and the hemodynamic study of Type-A dissection too is first study onto the best of our knowledge.

PUBLIC ABSTRACT

In this work, we have developed user interactive capabilities that allow us to perform segmentation and manipulation of patient-specific geometries required for Computational Fluid Dynamics (CFD) studies, entirely in image domain and within a single platform of ‘IAFEMesh’. Within this toolkit we have added commonly required manipulation capabilities for performing CFD on segmented objects by utilizing libraries like ITK, VTK and KWWidgets. With the advent of these capabilities we can now manipulate a single patient specific image into a set of possible cases we seek to study; which is difficult to do in commercially available software like VMTK, Slicer, MITK etc. due to their limited manipulation capabilities. Levelset representation of the manipulated geometries can be simulated in our flow solver (SCIMITAR-3D) without creating any surface or volumetric mesh. This image-levelset-flow framework offers few advantages. 1) We don’t need to deal with the problems associated with mesh quality, edge connectivity related to mesh models, 2) and manipulations like boolean operation result in smooth, physically realizable entities which is challenging in mesh domain. We have validated our image-levelset-flow setup with the known results from previous studies. We have modified the algorithm by Krissian et al. and implemented it for the segmentation of Type-A aortic dissection. Finally, we implemented these capabilities to study the hemodynamics in Type-A aortic dissection. Our image based framework is a first of its kind and the hemodynamic study of Type-A dissection too is first study onto the best of our knowledge.

TABLE OF CONTENTS

LIST OF TABLES	xi
LIST OF FIGURES	xii
CHAPTER 1. OVERVIEW	1
1.1 Introduction	1
1.2 Specific objectives.....	3
CHAPTER 2. PREVIOUS WORK IN MODELING PATIENT-SPECIFIC GEOMETRY	4
2.1 The need for patient-specific modeling.....	4
2.2 Steps involved in patient-specific modeling	6
2.2.1 Acquisition of patient-specific data from medical images.....	6
2.2.2 Image segmentation and image-based geometric modeling.....	11
2.2.3 Model preprocessing	17
2.2.4 Mesh generation	18
2.2.5 Assigning boundary conditions and computing flow	21
2.3 Current resources for patient-specific modeling	22
2.4 Our approach for image-based modeling	24
CHAPTER 3. BUILDING A USER INTERACTIVE TOOLKIT.....	26
3.1 Framework of IA-FEMesh toolkit.....	26
3.1.1 Visualization tool kit (VTK)	27
3.1.2 Insight segmentation and registration toolkit (ITK).....	27
3.1.3 KWWidgets.....	29
3.1.4 CMake	29
3.2 Targeted capabilities of IA-FEMesh	30
CHAPTER 4. IMAGE SEGMENTATION CAPABILITY.....	32
4.1 Image analysis	32
4.2 Objective 1. Build efficient image segmentation capability ..	33
4.2.1 The levelset methods for image segmentation	34
4.2.2 Levelset based two step segmentation to define embedded boundaries	35
4.2.3 Deriving levelset information to supply to the flow solver	37
4.3 Pipeline for general segmentation (Method I).....	38
4.3.1 CropImageFilter:	40
4.3.2 RescaleIntensityImageFilter:.....	40
4.3.3 SigmoidImageFilter:	41
4.3.4 ThresholdImageFilter:	42
4.3.5 FastMarchingImageFilter:	43
4.3.6 SignedMuarerDistanceMapperFilter:.....	44
4.3.7 GeodesicActiveContourLevelSetImageFilter:	44
4.4 Segmentation of aorta (Method II).....	46
4.5 Segmentation of aortic dissection (Method III)	47
4.5.1 Extraction of outer wall.....	49
4.5.2 Extraction of dissection wall.....	49
4.5.3 Segmentation of dissected aorta.....	50

4.6	Validation of Method I (Objective 1) using phantom model of ICA	51
4.7	Validation of Method II (region growing based segmentation).....	53
4.8	Validation of Method II and Method III using overlap fraction (OF)	53
4.9	Conclusions on image segmentation capabilities.....	55
CHAPTER 5.	FROM LEVELSETS TO FLOW COMPUTATIONS.....	56
5.1	Objective 2: Ensure image to flow computation.....	56
5.2	The levelset method	58
5.3	Deriving levelset information to supply to the flow solver.....	59
5.4	Flow solver with embedded boundaries.....	59
5.5	Cartesian grid method	59
5.6	Mesh structure	60
5.7	Solver validation	61
5.8	Validation of image-levelset-flow setup (flow in cylinder and intracranial aneurysm).....	62
5.9	Conclusions on from levelset to flow simulation.....	64
CHAPTER 6.	MANIPULATION OF SEGMENTED OBJECTS	65
6.1	Objective 3: Add general object manipulation capabilities ...	65
6.2	Conventional route of object manipulation.....	66
6.3	Marching cubes method	67
6.4	Object manipulation capabilities added in IAFEMesh.....	69
6.4.1	Center calculation.....	69
6.4.2	Extraction of oblique plane	71
6.4.3	Extrusion of vessel.....	71
6.4.4	Remove entities from object of interest	73
6.5	Conclusions on manipulation capabilities.....	75
CHAPTER 7.	EXAMPLES OF APPLICATIONS - HEMODYNAMICS STUDIES	76
7.1	Effect of the orientation of flow extension.....	76
7.2	Analysis of flow features in the diseased vessel with aneurysm and without aneurysm	77
7.3	Effect of root on hemodynamics and study of rupture risk in ascending aorta	79
7.3.1	Methods.....	80
7.3.2	Results.....	82
7.3.3	Discussions.....	85
7.3.4	Conclusions.....	86
7.4	Understanding hemodynamics in dissected aorta	86
7.4.1	Introduction and Motivation	86
7.4.2	Previous studies in literature	88
7.4.3	Material and methods.....	92
7.4.4	Results.....	95
7.4.5	Conclusions and limitations on hemodynamics in aortic dissection.....	101
7.5	Conclusions on examples of applications	102
CHAPTER 8.	SUMMARY	103
CHAPTER 9.	FIGURES	106

BIBLIOGRAPHY..... 162

LIST OF TABLES

Table 1	Comparison of Sigmoid-GAC based segmentation method (Method 1) with Method 0.....	52
Table 2	A Comparison of region based segmentation method with Method 0.....	53
Table 3	Overlap fraction (OF) between manually segmented and region based segmentation results for patient specific images of aortas P1, P2, D1 and D2. High overlap fraction is indicative of high spatial agreement between volumes.....	55
Table 4	List of number of cells in different cases simulated	82

LIST OF FIGURES

Figure 1	Geometry of Intra Cranial Aneurysm (ICA) showing its two inlets and two outlets and body or flow domain	106
Figure 2	Conventional route of Image based modeling	107
Figure 3	Our approach for image based modeling without body conforming mesh.....	108
Figure 4	Grid domain of (a) body fitted mesh and (b) non-body fitted mesh...	109
Figure 5	A schematic diagram of levelset field	109
Figure 6	Comparison of grid spacing for the case of flow past a cylinder in two dimensions with (a) body conforming mesh and (b) non-body conforming mesh.....	110
Figure 7	Current route of Image based modeling in our Thermo fluids group based on Active contour without edges.....	111
Figure 8	Steps involved in obtaining region of interest from original image in current set up of patient-specific modeling.....	112
Figure 9	Steps involved in image segmentation (a) image showing Region of Interest (ROI) (b) cropped ROI with pixels scaled from 0.0 to 1.0 colored by intensity (c) result of SigmoidImageFilter highlighting a window of pixel intensities of interest so that the region of interest becomes distinct (d) placing seeds(Red) for FastMarchingImageFilter (e) result of FastMarchingImageFilter used as initial guess for GeodesicActiveContourLevelSetImageFilter (f) zero levelset : result of GeodesicActiveContourLevelSetImageFilter	113
Figure 10	Collaboration diagram of GeodesicActiveContourLevelsetImageFilter applied to segmentation pipeline.....	114
Figure 11	Results of fast marching filter at time steps (a) 60 t, (b) 70 t, (c) 80 t and (d) 90t for Intra Cranial Aneurysm (ICA). User should aim at obtaining result of FastMarchingImageFilter as close to final segmentation result as possible	115
Figure 12	Model of aneurysm obtained from UIHC acquired by normal imaging procedures for aneurysms	116
Figure 13	Validation of pELAFINT Flow solver with results from paper by (Van De Vosse et al. 1989).....	117
Figure 14	Types of aortic dissections: DeBakey Type I (ascending, arch and descending aorta), Type II (ascending aorta only), Type III (descending aorta only). Stanford TypeA include Type I and II dissections.....	118

Figure 15	Dacron graft placed after surgery of Type-A dissection in (A) ascending aorta and aortic arch (B) only ascending aorta.....	118
Figure 16	Qualitative comparison of segmented result, (top) original phantom of ICA with dimensions, (bottom) segmented result of the phantom using segmentation Method I.....	119
Figure 17	Sixteen standard configurations of the Marching Cubes algorithm. The intersection of an isocontour with the edges of a computational cell leads to arbitrary polygonal shapes. Marching cubes efficiently determines the shape from a hexadecimal lookup table.....	120
Figure 18	Idealized models used as approximation of real geometries for numerical study of (a) bladder, (b) aneurysm and (c) coronary artery	121
Figure 19	Concept of zero levelset function where we have at (a) a circle representing original front which lies in X-Y plane and is propagating in directions shown by arrow, and at (b) levelset function in the form of cone surface where it shows position of zero levelset as the front propagates. Current position of the front is given by solid blue line and position of the front at different points in time are given by dotted lines	122
Figure 20	Flow chart of PaintImageGroup	123
Figure 21	Flowchart of EraseImagePixelGroup.....	124
Figure 22	Flowchart of segmentation of aorta or outer wall (Method II).....	125
Figure 23	Flowchart of segmentation of dissected aorta (Method III). Algorithm for detection of dissection wall is in inset on right.....	126
Figure 24	Flowchart of NormalExtrusionGroup.....	127
Figure 25	A comparison of results of (a) region growing based segmentation, (b) geodesic active contour segmentation with higher curvature value, and (c) geodesic active contour based segmentation with lower curvature value. The effect of parameters chosen in geodesic active contours for the same input obtained from region growing filter can be seen in (b) and (c)	128
Figure 26	Flowchart for region growing based segmentation of healthy aorta P1 and P2. Original CT scans are shown in level 1, result of preprocessing using region growing method is shown in level 2. Zero levelset obtained from geodesic active contour superimposed on original image is shown in level 3 and final extracted zero levelset is shown in level 4.....	129

Figure 27	Flowchart showing different stages of segmentation of dissected aortas D1 (top row) and D2 (bottom row). (a) CT scan of a patient with Type-A dissection; (b) image displaying segmented object of interest or outer wall of aorta; (c) zero levelset representation of the outer wall in green; (d) dissection wall in orange and (e) final extracted zero levelset representation of dissected or diseased aorta in white showing dissection wall in orange	130
Figure 28	A comparison of two models of ICA created by cutting cross section of inlets and outlets along the axes and extruding (white) Vs cutting the cross sections normal to the geometry and extruding the cross section (red)	131
Figure 29	Plots of velocity vectors scaled and colored by velocity magnitude obtained for Model 1 and Model 2, 3d reconstructions of ICA with flow extension normal to the boundaries and parallel to the axes respectively. The maximum velocity is seen at the left entrance to the aneurysm while the rest of the aneurysm has lower velocity and recirculation can be seen.....	132
Figure 30	A comparison of pressure in ICA for Model 1 (top) and Model 2 (bottom). The geometries are modeled with flow extension normal to the boundaries and parallel to the axes respectively. Inlet arteries have high pressure of 17 Pascals and outlets have 4 to 0 Pascals	133
Figure 31	A comparison of pressure distribution on the same cross-sections and velocity vectors colored by velocity magnitude in Model 1 and Model 2 in the study of effect of orientation of extrusions at inlet.....	134
Figure 32	Plot of pressure gradient in Pascals for study of vessel with aneurysm (top) Vs without aneurysm (bottom) under same boundary condition	135
Figure 33	A comparison of velocity vectors passing through a slice, colored by pressure gradient) in the middle of aneurysm and the velocity vectors passing though slice (colored with pressure gradient) at the same position in the model without aneurysm. The insets in the left of each models show the slice and velocity vectors individually	136
Figure 34	Major steps in segmentation of dissection wall and final dissected aorta (true and false lumen) are shown using a representative slice. 1) Original image, 2) binary mask created from segmenattion of outer wall, 3) Original pixels are extracted by applying mak B1 on the original image, 4) labeled image obtained by assigning a label FL to pixels below 'm' and TL to those above 'm'and applying median filter, 5) output of gradient magnitude filter where the boundary pixels of outer wall have higher intensity and boundary pixels separating true and false lumen have lower intensity, 6) pixels belonging to dissection wall DW are extracted, 7) resulting binary image obtained by removing pixels belonging to DW from B1	137

Figure 35	Result of segmentation of dissected aorta in IAFEMesh utilizing region growing, morphologic and levelset filters. The green region is true lumen and the pink is false lumen. The right image is dissected wall	138
Figure 36	Models created for analyzing the effect of excluding head root and tricuspid valve in hemodynamic analysis. Models of Case A, Case B and Case C, represent the geometry at peak of systole with sinuses and valve (top left), peak of systole without sinuses and valve (top right) and early diastole (bottom) . Inserts at the top left of both models of systole show a transverse view of the sinuses and fully open valve forming a triangular inlet to the sinuses for Case A, as opposed to a circular cross section for Case B, without the valve. Insert at the top left of Case C shows the transverse view of the sinuses with coronaries and fully closed aortic valve. Inserts at the bottom left of Case A and C show fully open and completely closed positions of the leaflets of the aortic valve during peak systole and early diastole.....	139
Figure 37	Streamlines colored by pressure for peak systole with sinus and valve (top left), peak systole without sinus and valve (top right), early diastole with sinus and valves (bottom). In Case A, high velocity jet of blood is seen passing through the tricuspid valve aperture directed towards the arch while the blood directed towards the cusps of the sinus is clearly seen decelerated and recirculating. This phenomenon is characteristic of systole and is seen missing in Case B which stresses the importance of including valve and sinus to capture more realistic flow. In Case C, blood flowing with relatively higher velocity and pressure in the arch is seen to gradually decelerate recirculate as it approaches the sinus with valves completely closed. In the insets to the left of Cases A and C, a representative cross section of sinus of valsava showing pressure distribution in: peak systole with sinus and valve (left) and early diastole with sinus and valve (right) are shown. In both the cases pressure is higher in non-coronary sinus. This clearly shows that out of the three coronaries pressure is highest in the non-coronary sinus throughout the cardiac cycle.....	140
Figure 38	Plots of velocity distribution in the coronal plane during peak systole with sinus and valve (top left), peak systole without sinus and valve (top right), and early diastole with sinus and valve (bottom). The effect of incorporation of sinus and valve aperture is clear in figure of Case A as the high velocity jet adhered to the inner wall is seen directed towards the arch while the blood directed towards the cusps of the sinus is seen decelerated and recirculating. This phenomenon is characteristic of systole and is seen missing in Case B which stresses the importance of including valve and sinus to capture realistic flow	141

Figure 39	Wall shear stress distribution during peak systole with sinus and valve (top left), peak systole without sinus and valve (top right), and early diastole (bottom). WSS is clearly higher in the inner walls in peak systole as seen in both Case A and B, but their magnitudes and the region of high WSS is more distributed for Case A due to the high velocity jet. Relatively low WSS is seen during early diastole, Case C, due to comparatively lower velocity..	142
Figure 40	Validation of Image to flow set up by comparing entrance length obtained via simulation against empirical results at Re 50. Binary image of the cylinder (top), rank distribution of flow domain (second), velocity vectors in 3D colored and scaled by velocity magnitude at different locations along the cylindrical domain (third), 2d view of ZY plane colored by velocity magnitude (fourth), velocity vectors at different locations in ZY plane showing fully developed flow at 5 m distance downstream (bottom) which agrees with the empirical result for pipe flow when diameter is 2m and Re is 50.....	143
Figure 41	Validation of flow results obtained from image-to-flow set up against flow simulation obtained from FLUENT v14.5.0 (ANSYS Inc., Canonsburg, PA) for intra cranial aneurysm (ICA) for the same boundary conditions of 0.36m/s at left inlet, 0.37m/s at right and 50:50 mass split at the two outlets. Pressure distribution obtained of surface of ICA from in-house flow solver (top) and FLUENT v14.5.0 (ANSYS Inc., Canonsburg, PA) (bottom) are in agreement	144
Figure 42	Validation of flow results obtained from image-to-flow set up against flow simulation obtained from FLUENT v14.5.0 (ANSYS Inc., Canonsburg, PA) for intra cranial aneurysm (ICA) for the same boundary conditions. Stream ribbons of velocity vectors colored by velocity magnitude from in-house flow solver (top-left) and ANSYS FLUENT v14.5.0 (ANSYS Inc., Canonsburg, PA) (bottom-left) are shown in this figure. Similarly, corresponding plots of wall shear stress are shown in right side of this figure. Insets point at the regions where results from in house solver appear slightly higher compared to FLUENT v14.5.0 (ANSYS Inc., Canonsburg, PA).....	145
Figure 43	(a) CT scan of a patient with Type-A dissection; (b) visualization of true lumen (green), false lumen (pink) and dissection wall with patched tears of the dissected aorta; (c) reconstructed Diseased model with insets highlighting the location of tears and (d) Baseline aortic model created by merging true lumen, false lumen and dissection wall with tears patched, representing healthy aorta used in Study 1	146

Figure 44	<p>Visualization of velocity (top) and pressure (bottom) in diseased model with Type-A dissection and baseline model representing healthy aorta at peak systole. Boundary conditions applied at different inlets and outlets are shown in left. Pressure at Peak systole for normal aorta model (top-left), diseased before surgery case (top-right), after-surgery case (bottom-left), and after-surgery true lumen expanded case (bottom-right). Total pressure in ascending aorta becomes higher in before-surgery case than in normal model due to the decrease in cross sectional area for the same amount of mass influx.. Since the cross sectional area for mass flow decreases further in after-surgery model due to the closure of entry point of false lumen, the pressure at the ascending aorta escalates further</p>	147
Figure 45	<p>Visualization of Time Averaged Wall Shear Stress (AWSS) for baseline undissected (top) and diseased model (bottom). Comparison of both the views shown in left and right show the presence of higher AWSS in cross sections with minimal cross section (left) and entry point of primary tear (right) in diseased case than in baseline case</p>	148
Figure 46	<p>Model of Type-A aortic dissection (left) with 5 tears and Repair-I (right most). The first tear is at the junction of ascending aorta and arch, the location is highlighted in inset in the middle. This tear is removed in the after surgery model as shown in inset. Second tear is at the middle of descending aorta and exists in both before surgery and after surgery models. It's magnified view is shown in inset. Three tears are present at the bottom of descending aorta which exist in both before and after surgery models and the magnified view is shown in the inset in the middle</p>	149
Figure 47	<p>Models of Diseased (left) and Repair-II (right) cases highlighting the differences in the insets. Primary and secondary tears present in original Diseased model are patched in Repair-II model which can be seen in insets in first and second row. Three tertiary tears present in Diseased case are kept intact in Repair-II model as shown in inset in third row</p>	150
Figure 48	<p>A comparison of velocity vectors of dissection models- baseline, Diseased, Repair-I and Repair-II from left to right, colored and scaled by velocity magnitude at peak systole (top), mid systole (middle) and early diastole (bottom). Exact point in cardiac cycle captured in the images have been highlighted by red dot in the mass flow Vs time plot in the right. Flow in the false lumen decreases throughout the cardiac cycle in both repair cases when compared to Diseased case. However, flow is localized to bottom tertiary tears in Repair-II while in the region extending from the secondary tear to tertiary tears in Repair-I.....</p>	151

Figure 49	Top view of velocity vectors of dissection models- baseline, Diseased, Repair-I and Repair-II from left to right, colored and scaled by velocity magnitude at peak systole (top), mid systole (middle) and early diastole (bottom). Exact point in cardiac cycle captured in the images have been highlighted by red dot in the mass flow Vs time plot in the right. Flow in the false lumen decreases throughout the cardiac cycle in both repair cases when compared to Diseased case. However, flow is localized to bottom tertiary tears in Repair-II while in the region extending from the secondary tear to tertiary tears in Repair-I.....	152
Figure 50	A comparison of velocity vectors at same cross sections for Diseased, Repair-I and Repair-II case to check the effectiveness of Repair-I model in reducing mass flux in false lumen.....	153
Figure 51	Mass flow rate throughout a cardiac cycle in true lumen (right) and false lumen (left) at different cross sections with respect to time for baseline (true lumen only), Diseased, Repair-I and Repair-II cases. The cross sections analyzed are shown in the middle of the figure....	154
Figure 52	A comparison of time averaged wall shear stress (AWSS) for diseased (top), repair-I (middle) and repair-II (bottom). Views of true lumen and false lumen of the same model are shown in left and right. Higher AWSS is seen in true lumen mainly in the collapsed region as the flow is restored after repair in the two repair models. Similarly, low AWSS is seen in the false lumen at the arch region due to the patching of primary tear in the repair models.....	155
Figure 53	Models of diseased cases where the location of primary tear are altered to understand the role of location of tear in the mass flow in aortic arch. Model with primary tear located at ascending aorta (top-left), model with primary tear located before left-subclavian artery in arch region (top-right), model with primary tear located in descending aorta after left left-subclavian artery (bottom-left), model with primary tear located at the middle of descending aorta (bottom-right) are shown in this figure and they are addressed as Tear-Loc1, Tear-Loc2, Tear-Loc3, and Tear-Loc4 (or Repair-I) respectively	156
Figure 54	Visualization of velocity vectors at peak systole (top row), mid systole (middle row), and early diastole (bottom row) for dissected models with varied location of primary tear namely, TearLoc1, TearLoc2, TearLoc3, and TearLoc4	157
Figure 55	Top view of velocity vectors at peak systole (top row), mid systole (middle row), and early diastole (bottom row) for dissected models with varied location of primary tear namely, TearLoc1, TearLoc2, TearLoc3, and TearLoc4.....	158

Figure 56	Mass flow rate in true lumen (right) and false lumen (left) at different cross sections with respect to time for models of dissected cases-with primary tear at varying locations, namely, Tear-Loc1, Tear-Loc2, Tear-Loc3, and Tear-Loc4 (or Repair-I). The cross sections analyzed are shown in the middle of the figure ...	159
Figure 57	Variation of pressure in primary tear in a cardiac cycle for dissection models with varying location of primary tear, namely, Tear-Loc1, Tear-Loc2, Tear-Loc3, and Tear-Loc4.....	160
Figure 58	Grid independence study was carried out for normal undissected aorta where three models coarse, medium and refine were generated with average element of size 0.002 mm, 0.001 mm and 0.00075 mm respectively. Velocity magnitudes for all the models at six cross sections as shown (middle) were analyzed. Three of the six cross sections are shown as slice1, slice 2 and slice 3 on right. Percentage error (left) were calculated for these cross sections with respect to the refine case. Medium case with average element size of 0.001 mm was selected for this study	161
Figure 59	Flowchart for ErasePixelImageGroup	162

CHAPTER 1.OVERVIEW

1.1 Introduction

The conventional route for patient-specific modeling can be summarized as the following set of steps: 1) acquire data in the form of images from different imaging modalities such as CT, MRI or Ultrasound, 2) *hand-segment* ,delineating features of interest, the images to segregate the subject of interest, and 3) use the segmentation in a separate commercial package such as Gridgen, GAMBIT, etc. to construct a volumetric mesh for the purpose of CFD analysis (See Figure 2). Many sophisticated packages such as Vascular Modeling Toolkit (VMTK) (Antiga et al. 2008; Antiga and Steinman 2006), 3DSlicer (Steve and Ron; Steve et al. 2006), Insight Registration and Segmentation Toolkit (ITK), Matlab have been developed to handle the components of this procedure in an accurate, robust, and efficient manner. Unfortunately, these packages have limited capabilities to perform modifications to the geometries represented in images which make them sub-optimal for applications such as virtual surgical planning, where the user needs to manipulate a set of images into a new set of possible cases.

A normal procedure of creating an input levelset field, a signed distance field (Sethian 1999), for patient-specific geometries for Parallel Eulerian Lagrangian Algorithm for Interfacial Transport (pELAFINT) would previously follow following paths (See Figure 3 and Figure 8). After obtaining a crude image, the image would be down sampled in Matlab so that it will be easier to trace the region of interest (ROI) in Tecplot. The original image would then be cropped in Matlab as per ROI. Such image would then be subjected to preprocessing steps like anisotropic diffusion, thresholding, labeling and then finally would be segmented using active contour based algorithm followed by levelset field generation by implementing fast marching algorithm. The entire steps of filtering or removing noise to enhance the ROI from the image (Glasbey and Horgan 1995) and obtaining a levelset are coded in Fortran 90 without graphical user interface (GUI). It is hence not user friendly and requires technical expertise. In order to view intermediate results one would need to use different visualization software like Tecplot or Paraview. So in summary,

current method of obtaining levelset of an image would require a user to simultaneously use Matlab, Fortran code and Tecplot. This process is definitely possible but not ideal from an end-user perspective. Involvement of numerous software also poses challenges in parallelizing the segmentation process, which is a significant setback since parallelization of all the steps involved in image segmentation to flow calculation is one of our important goals.

Further, optimal values of scaling parameters of segmenting *filters*, operations to accentuate features of interest (Glasbey and Horgan 1995), can only be determined by experimentation. And in a set up like this, where there is no user interface and one needs to juggle between different software, fine tuning of parameters is definitely possible but not economical in terms of time and effort. This restricts the whole process as an art rather than science.

Furthermore, for research purposes geometries often need to be modified to study the isolated effect of different parameters. For instance, if the goal is to analyze the effect of branch flow in patient-specific arteries then one would desire software capable of adding or removing branches from patient-specific arteries. Similarly, if the goal is to study the effect of varying sizes of aneurysm in patient-specific geometries, one would look for software capable of adding or removing aneurysm or altering its size. There are software like Pro/ENGINEER, AutoCAD, Gambit, Pointwise capable of creating and manipulating surfaces and volumes for ideal geometries. However, these software are not properly devised to manipulate patient-specific geometries. So, if such patient-specific geometries need to be manipulated then one would require knowing little tricks of various software and carve one's methodology through it. Again, it is definitely achievable but with a lot of pain to the user. Hence, in order to manipulate geometries for patient-specific studies, we aimed to build a customized package that contains user-friendly tools for segmentation (tools that are tailored for our purpose) like in Slicer3D and would also contain geometry manipulation capabilities like in solid modeling software like AutoCAD and would allow control over sensitive parameters along visualization at every step. This package built using the libraries of ITK, VTK and KWWidget will not only decrease the chaos a user would otherwise experience while working between many software and file types but would also add a

nice GUI to our lab's segmenting capabilities and would increase efficiency and effectiveness of the whole procedure. Our aim to make it as user-friendly and user-interactive as possible, add different types of segmenting and geometry manipulation capabilities as seen necessary to the image/geometries targeted for research.

As mentioned earlier, optimal values of scaling parameters can only be determined by experimentation. Obtaining such optimum values becomes easier if there is an interactive mechanism by which the user can supervise and adjust the effects of each parameter at every step of filtering.

In this work we aimed to develop IA-FEMesh (Iowa FE Mesh), a toolkit that has proved its merit in skeletal modeling, as a complete package that can take care of all needs related to image segmentation and object manipulation for modeling biological fluid flow. The attempt is to make this experience with IA-FEMesh as user interactive as possible so that the user will have more knowledge about the process. Such interaction may not be of great importance to people with imaging majors but for those with none it is of utmost importance.

1.2 Specific objectives

Following are the specific objectives of this thesis.

1. The first aim was to build a user interactive toolkit that has the capability of segmenting an image and obtaining a levelset field of the object of interest.
2. The second aim was to verify that the process of image segmentation to flow calculation can be executed without the need of complex volume mesh generation with the help of our toolkit and in-house flow solver.
3. The third objective was to enable (within the toolkit) commonly required manipulation of the segmented objects.

CHAPTER 2. PREVIOUS WORK IN MODELING PATIENT-SPECIFIC GEOMETRY

Advancement in numerical methods, high speed computational capacity and three-dimensional imaging techniques have enabled quantification of patient-specific physiologic models (Taylor and Figueroa 2009). Patient-specific models are not only being used to test hypothesis related to diseases and to analyze the effectiveness of implants, medical devices in the field of predictive medicine but they are also being widely used guide cell culture, and perform animal experiments etc. (Taylor and Figueroa 2009). This work is an application of patient-specific models to analyze the hemodynamics using CFD.

2.1 The need for patient-specific modeling

Realistic modeling of fluid flows in the human body imposes many challenges. Accurate representation of irregular and tortuous geometries acting as a conduit for biological fluid flow, determination of appropriate boundary conditions which generally requires invasive measures are two of the major hurdles laid in this path. Traditionally researchers used to overcome the challenges of modeling complicated real geometries by simplifying them and making their idealized representation by approximating anatomical features as closely as possible. Examples include, representation of human aortic arch and coronary arteries with semi-circular tube of certain mean diameter measured from cadavers (Santamarina et al. 1998) (Figure 18 c); use of concave arc of some radius for approximation of atherosclerosis in an artery which is again modeled after a straight tube of certain radius (Chandran, Yoganathan, and Rittgers 2012); spheroidal representation of bladder (Beek 1997) (Figure 18 a), spheroidal representations of cerebral aneurysms in which metrics like bulb size and neck diameter are varied to predict conditions that may lead to aneurysm growth (György Paál 2007) (Figure 18 b); bifurcating pipe models for studying the effects of flow conditions on aerosolized drug delivery mechanisms in the trachea-bronchial airway (Luo et al. 2004) or on the hemodynamics of a coronary arterial bypass graft (Kute and Vorp 2001); or the effects of geometry and motion on the rate of nutrient mixing at the antro-duodenal junction (S. Dillard, Krishnan, and Udaykumar H S 2007).

Often simple models are found useful in answering questions about significance of geometric parameters (Pekkan et al. 2005). They help by secluding the effects caused by geometry and enabling the sole analysis of physical trends they cause. It is easier to manipulate such regular geometries in terms of size and position and examine the effect of geometric variation on flow fields (S. Dillard et al., 2014).

Simplification of real geometries lead to the loss of local geometric details which are often important in picturing real flow conditions and can be crucial when accuracy is of concern (Xiong et al. 2011). For example, in the study conducted by Cezeaux and Grondelle in 1997, it was found that the tubular representation of coronary arteries fail to incorporate the tapering effect and hence the hemodynamics observed were far from accurate; flow features and stress parameters predicted with the aid of simplified airway models in the trachea were not found to be adequate to simulate the real picture (Lin et al. 2007); similar observations were also made in the case of intracranial aneurysms (ICAs), hence real geometric modeling could actually serve as a valuable factor in predicting aneurysmal rupture risk (B. Ma, Harbaugh, and Raghavan 2004; J. R. Cebal et al. 2011) or local hemodynamics in aortic arch or state of flow past atherosclerosis (Chandran, Yoganathan, and Rittgers 2012).

Local vascular geometry is one of the most crucial deciding factors in hemodynamics (S. Dillard et al., 2014). These vascular geometries are subject-specific. Though they are similar among individuals they essentially vary anatomically (Xiong et al. 2011), and hence the local hemodynamics vary too. Because of afore mentioned reasons majority of researchers today are using image based, patient-specific models (Steinman and Taylor 2005; Taylor and Figueroa 2009). Further, surgical practitioners consider patient-specific models to be valuable because of the information they decipher when planning surgical procedures (Cebal et al. 2011; Sforza, Putman, and Cebal 2012; Sundareswaran et al. 2009; Mantha et al. 2006). Similarly, medical devices industries believe patient-specific models can play key part in the development of these devices because development of such devices require specific information about the disease to be cured, location of the disease and where the device needs to be deployed etc. (Taylor and Figueroa

2009). Furthermore, patient-specific models can be studied to recognize common hemodynamic features and variations across ranges of patients and draw statistically significant conclusions (Taylor and Figueroa 2009).

There have been major advancements in the imaging sector in past few years particularly in the fields of computed tomography (CT), magnetic resonance (MR), Ultrasound, and transesophageal echocardiography (TEE), computing techniques, storage capabilities, and hardware too have advanced in parallel. Together, these developments have made it possible to create realistic 3D patient-specific models derived from images. It is hence feasible to model the detailed complexities associated with patient-specific geometries and obtain hemodynamics predictions that are closer to reality.

2.2 Steps involved in patient-specific modeling

In general, patient-specific modeling for flow computations comprises of following steps (See Figure 2) which will be dealt in detail in the subsequent portions of the section.

- a) Acquisition of patient-specific data
- b) Segmentation of region of interest
- c) Model preprocessing
- d) Discretization (Typically mesh generation) of the segmented object
- e) Assigning boundary conditions and Computing flow

2.2.1 Acquisition of patient-specific data from medical images:

Medical images can be used for diagnosis and staging of diseases. For instance, imaging can be used to find out where a cancer is located in the body, if it has spread, and how much is present. Used in this way, imaging can help determine what stage (how advanced) the cancer is, and if the cancer is in, around, or near important organs and blood vessels. Imaging can be used to make cancer treatments less invasive by narrowly focusing treatments on the tumors. For instance, ultrasound, MRI, or CT scans may be used to determine exact tumor locations so that therapy procedures can be focused on the tumor, minimizing damage to surrounding tissue.

Biomedical imaging technologies utilize different modalities like x-rays (CT scans), sound (ultrasound), magnetism (MRI), radioactive pharmaceuticals (nuclear medicine: SPECT, PET) or light (endoscopy, OCT) to assess the current condition of organ or tissue and monitor a patient condition. For CT, scan time takes around 30 seconds while for MRI it is about 30 minutes. CT Scanning is useful for screening diseases such as colon cancer, detecting injuries and abnormalities in the head, chest, heart, abdomen and extremities. Ultrasound applications are used for diagnostic applications such as visualizing muscles, tendons, internal organs, to determine its size, structures, any lesions or other abnormalities.

2.2.1.1 CT:

Biomedical imaging began with the invention of first x-ray in 1895. Ever since then, x-rays have come a long way. With the use of solid-state electronics the exposure time has now reduced to milliseconds hence drastically decreasing the radiation dose. Usage of contrast medium made visualization of organs and blood vessels possible. Digital imaging gave rise to Computed Tomography (CT) scanners which made it possible to watch real-time x-rays on monitor, a process popularly known as x-ray fluoroscopy – to guide invasive procedures like angiograms and biopsies. Typical radiation dose from CT ranges from 2 to 10 millisievert (mSv) which is about the same amount that an individual would receive from background radiation. CT is used to trace bone injuries, chest and lung imaging and cancer detection. MRI is more versatile than CT. It is mostly suited for soft tissue evaluation like ligament and tendon injury, spinal cord injury, brain tumors etc., can be used to examine large variety of medical conditions. It is good for obtaining soft tissue differentiation especially with intravenous contrast. Higher imaging resolution and less motion artifact due to fast imaging speed. Due to the high contrast resolution of CT scanning, differences between the tissues are more apparent compared to other techniques. Further, imaging in CT scan removes the superimposition of structures other than the area of interest.

2.2.1.2 Ultrasound:

Ultrasound machines are portable and are faster in capturing data than other modalities. It does not emit radiation and hence it holds great possibility of obtaining real time images. However, in compare to other modalities, ultrasound produces images of low visual quality requiring more robust denoising algorithms. Ultrasounds are usually not used for bony structures. Instead they are used for internal organs of the body. In an ultrasound, a high frequency sound wave is produced in short pulses at the desired frequency and focused at the region of interest in the body. These sound waves are partially reflected back from the body, received by a transducer and sent to the ultrasonic scanner, where they are processed and transformed into a digital image. The formation of the image depends on the time and the strength of the echo, and is displayed on the computer screen for analysis.

Intravascular Ultrasound consists of ultrasound tipped catheter which is protruded inside an artery to obtain its cross sectional information. It is very popular in case of relatively straight arteries like carotid, femoral but when working with coronaries, it fails to give back information about the path followed. To overcome this shortcoming IVUS is often used in combination with angiogram. By combining IVUS information with 2d planar information of angiogram even the tortuous coronaries can be imaged with great precision (Shrestha 2010).

2.2.1.3 MRI:

In 1977, the first MRI exam was performed on a human being. The procedure was long and complicated, taking over 5 hours to produce a single image. Although researchers had struggled for over 7 years to reach that point, by today's standards, the image was unrefined and coarse. Critical advances in technology development now allow physicians to image in seconds what used to take hours. MRI has been used successfully for over 15 years to generate soft tissue images of the human body, making it the technique of choice for the routine diagnosis of many diseases and disease processes. Today more than 60 million noninvasive, diagnostic MRI procedures are performed worldwide each year. It is able to demonstrate subtle differences

between the different kinds of soft tissues. One of the greatest advantages of MRI is the ability to change the contrast of the images. Small changes in the radio waves and the magnetic fields can completely change the contrast of the image. Different contrast settings will highlight different types of tissue.

Using a very powerful magnet and pulsing radio waves the detection coils in the MRI scanner read the energy produced by water molecules as they mis-align themselves after each RF alignment pulse. The collected data is reconstructed into a two dimensional illustration through any axis of the body. Bones are virtually void of water and therefore do not generate any image data. This leaves a black area in the images.

2.2.1.4 Nuclear imaging (PET and SPECT):

Nuclear imaging uses low doses of radioactive substances linked to compounds used by the body's cells or compounds that attach to tumor cells. Using special detection equipment, the radioactive substances can be traced in the body to see where and when they concentrate. Two major instruments of nuclear imaging used for cancer imaging are positron emission tomography (PET) and single-photon emission computed tomography (SPECT) scanners.

The PET scan creates computerized images of chemical changes, such as sugar metabolism, that take place in tissue. Typically, the patient is given an injection of a substance that consists of a combination of a sugar and a small amount of radioactively labeled sugar. The radioactive sugar can help in locating a tumor, because cancer cells take up or absorb sugar more avidly than other tissues in the body.

2.2.1.5 PET:

PET scans may play a role in determining whether a mass is cancerous. However, PET scans are more accurate in detecting larger and more aggressive tumors than they are in locating tumors that are smaller than 8 mm and/or less aggressive. They may also detect cancer when other imaging techniques show normal results. PET scans may be helpful in evaluating and staging

recurrent disease (cancer that has come back). PET scans are beginning to be used to check if a treatment is working and if a tumor cells are dying and thus using less sugar.

2.2.1.6 SPECT:

Similar to PET, single photon emission computed tomography (SPECT) uses radioactive tracers and a scanner to record data that a computer constructs into two or three-dimensional images. A small amount of a radioactive drug is injected into a vein and a scanner is used to make detailed images of areas inside the body where the radioactive material is taken up by the cells. SPECT can give information about blood flow to tissues and chemical reactions (metabolism) in the body.

2.2.1.7 Digital subtraction angiography (DSA):

If two images are taken of an object one in a normal mode and the other one with enhanced contrast for the subject of interest, then the subtraction of the two digital images can give excellent quality results. This is a process similar to masking in image segmentation but with much higher resolution and contrast. After taking the first radiograph (mask image), the person is injected with contrast agent which is designed to elicit blood, and second radiograph (live image) is taken and stored in computer. The digital images of these two radiographs are then subtracted and their difference is amplified.

2.2.1.8 Endoscopy:

Endoscopy is generally used to confirm or know the degree of disease when other imaging modalities are found unsuitable for the case. In endoscopy, endoscope having a camera and light is inserted inside the body and the entire travel video is recorded. It is considered minimally invasive procedure of imaging.

Different modalities of biomedical imaging are useful for different purposes. For image acquisition of vascular anatomy related to cardiovascular mechanics non-invasive methods like CT, MRI, 3D Ultrasound (3DUS) and Intravascular Ultrasound (IVUS) are popular due to the

sensitivity of the region being imaged. Contrast-enhanced CT and MRI are particularly popular for generating high-resolution volumetric images of many parts of the vascular tree. Generally, iodinated contrast is used in CT angiography, and a gadolinium-based contrast agent is used in magnetic resonance (MR) angiography. MRI has the additional advantage of being able to quantify physiologic parameters, including blood flow, wall motion, and blood oxygenation. CT is the best bet for checking damage in bone, Ultrasound for soft tissues.

2.2.2 Image segmentation and image-based geometric modeling

Image segmentation is a process of tracing different entities present in an image and distinguishing them from one another. It helps in visualizing medical data and diagnosing various diseases. Modern segmentation methods in the literature treat an image and its pixel values as a topological field of sorts, on which segmentation can be accomplished by computing pixel gradients or by minimizing a functional over the pixel field in a manner analogous to energy minimization. Segmentation techniques can be broadly classified into two: low level segmentation techniques that depend on per pixel information such as image intensity (e.g., thresholding and region growing methods) and high level segmentation techniques that depend on the spatial distribution of data such as image features (e.g., levelset based methods). Below is the discussion of some of the widely used methods of segmentation.

2.2.2.1 Threshold segmentation

Thresholding in its simplest form attempts to group image in to two broad categories based on a value taken as threshold. Hence, only 2 channels are generated and it cannot be used to multichannel the images. These methods are found useful for cases consisting of contrasted objects on a realm of consistent background like for distinguishing printed characters or blood cells etc. (Sonka, Hlavac, and Boyle 2008). Since real images have different levels of noise and the object of interest does not always have same intensity throughout the image, thresholding cannot be used for complete segmentation of the image however it is generally found useful for pre and post processing of such images.

If R be an image with finite sets of regions R_1, R_2, \dots, R_s ,

$$R = \sum_{i=1}^s R_i; R_i \cap R_j = \emptyset, i \neq j \quad (1)$$

Thresholding of image R with respect to intensity T would result in a binary image g such that:

$$g(i, j) = 1 \text{ for } R(i, j) \geq T, \quad (2)$$

$$= 0 \text{ for } R(i, j) < T.$$

Where $g(i, j) = 1$ is the value that the object of interest gets as foreground pixels and $g(i, j) = 0$ is the value for background pixel intensity for all the regions that were below the threshold.

2.2.2.2 Edge-based segmentation

Edge-based segmentation are one the earlier segmentation approaches that take advantage of information about the edges in an image which are determined by edge detecting operators, where edges represent the locations of discontinuities in image intensity. Canny edge detector, Hough transform (if object of interest are of regular shapes) are some of the popular and robust edge detecting algorithms. However, an image resulting from edge detector cannot be directly used as a segmented image because the resulting image could have close, partial or extraneous borders around the regions of interest. Hence, further post processing like edge linking may be required.

More the prior information about the boundaries is available better the segmentation result will be as there will be enough basis for correction of segmentation result. These methods are sensitive to noise specially those resulting in edges where no edges are present and those that dilute or erase the borders.

This method could give powerful segmentation in case optimality criterion is defined. Graph searching (Heuristic) or dynamic programming could then be employed for searching optimal paths, the choice of either depends on the calculation of the local cost function as well as evaluation function and on the quality of heuristics for an A-algorithm.

2.2.2.3 Region-based segmentation

It is a method of extracting image regions that are connected based on some predefined criteria like intensity and/or edge on image. It requires user interaction in the form of seed placement, after which all the pixels connected to that initial seed/seeds are compared based on some pre-defined criteria and those meeting the criteria gets extracted (Luis et al. 2005). It is generally used for delineating small objects in images like tumor, lesions etc. (Pham, Xu, and Prince 2000). The choice of the homogeneity criteria is hence critical for the success of this method. Such criteria generally depend on the image being segmented, noise involved and the object of interest. Based on those criteria and the type of connectivity used to determine the neighboring pixels and the strategy used to visit those pixels, one can find varied implementations of the region growing algorithms. The variations to region based approaches are region growing, region splitting and the combination of both split-merging.

2.2.2.4 Segmentation based on watershed transform

Watershed algorithm is thought of as one of the most reliable automatic and unsupervised algorithms developed for segmentation. The most alluring feature of this algorithm is its ability to adapt itself to different types of images and that it gives continuous boundaries. It functions by dividing the pixels into regions (catchment basins) by associating point in the image domain within local extrema in some feature measurement such as gradient magnitude or curvature. The main idea is: If a two dimensional image with gray scale intensity is considered, then the topology is formed such that the minima form the trough of basin and the maxima the crests. Let us assume there to be holes at the local troughs. Now, if water is released from the holes and allowed to fill the topology at a uniform rate, there will come a point when water levels in neighboring catchment

basins will start to merge. A dam is built to prevent merging at this point of time. In this manner, dam boundaries representing different regions of image with homogeneous intensities (Vincent and Soille 1991) will be obtained. It can hence be thought of as an algorithm that is utilizing the aspects of region based as well as edge based segmentation.

An image to which watershed algorithm can be applied should have single grayscale minimum at each region and the boundary should have grayscale maxima. Real images generally do not satisfy these criteria hence they are preprocessed to reduce the number of local minima and enhance the boundary. This method is highly sensitive to noise and contrast leading to over segmentation, however it fails to detect thin noise (Grau et al. 2004). It also requires smoothing because of poor localization of boundaries (Beucher, Yu, and Bileadu 1990).

2.2.2.5 Segmentation based on deformable models

Deformable models are popular approaches when it comes to medical image analysis (Hegadi, Kop, and Hangarge 2010; Tim McInerney and Terzopoulos 1996) in segmenting complex anatomies (Z. Ma et al. 2010) and in particular for segmentation of vascular structures (Hernandez et al. 2003). An overview of the impact of these models in medical image analysis can be found in a review paper by Tim McInerney and Terzopoulos in 1996. Deformable models are typically closed curves in 2D or surfaces in 3D, which deform under the influence of internal or external forces. The deformation of a model can be described in a Lagrangian fashion in which material points, a suitable parameterization of the model, are followed during their evolution. This technique is commonly referred to as *parametric deformable models*, *active contours* or *snakes*. On the other hand, such deformation model can also be described in Eulerian fashion, in which the motion of the model is observed through the evolution of scalar field values at fixed points in space (L. Antiga 2002). This second technique is a part of implicit deformable models popularly known as *levelset* approach (Sethian 1999). Mathematical foundations on which deformable models are built are beautiful amalgamation of geometry, physics and approximation theory; where geometry

represents the shape of the object, physics imposes constraints and approximation theories are responsible for fitting models to measured data.

2.2.2.6 Parametric deformable models

Parametric deformable models are those that represent curves and surfaces explicitly during deformation. These models, in their parametric forms can have formulation based on energy minimization or dynamic force formulation (Hegadi, Kop, and Hangarge 2010). Though both the formulations lead to same results, the solution of the first formulation satisfies minimum energy principle while the second one allows using general types of external forces (even negative gradient forces). The energy minimizing formulation is based on determining a parameterized curve that minimizes the weighted sum of internal energy and potential energy. The internal energy denotes tension or the smoothness of the contour and the potential energy is defined over the image domain and its value is minimum at object boundaries.

The dynamic formulation is one in which the model is formulated based on dynamic problem using the force formulation in contrary to adding an artificial time ‘t’ element in the static problem as done in energy minimizing formulation. Parametric deformable models have been extensively implemented in literature in the forms of dynamic programming (Terzopoulos, Witkin, and Kass 1988), greedy algorithm (A. A. Amini 1990) etc., and various improvements have been proposed, however reparameterization and topological adaptation still remain major limitations of these models and the main reason behind the popularity of geometric deformable models.

2.2.2.7 Geometric deformable models or levelsets

Levelset techniques are powerful mathematical tools to deal with various applications in imaging and computer vision. The main advantage of using levelsets is that arbitrarily complex shapes can be modeled and topological changes such as merging and splitting are handled implicitly. Levelset models are topologically flexible and can split and merge as necessary in the course of deformation without the need for re-parameterization. Using Osher and Sethian’s levelset approach, complex curves can be detected and tracked and topological changes for the evolving

curves are naturally managed. Levelsets can be used for image segmentation by using image-based features such as mean intensity, gradient and edges in the governing differential equation. In this method, a contour is initialized by a user and is then evolved until it fits the form of an anatomical structure in the image.

Levelset segmentation is a numerical method for tracking the evolution of contours and surfaces by embedding the contour as the zero levelset of a higher dimensional function called as levelset function $\psi(x, t)$. Levelset filters implemented make use of following general levelset equation:

$$\frac{\partial \psi}{\partial t} = -\alpha A(x) \cdot \nabla \psi - \beta B(x) |\nabla \psi| + \gamma Z(x) k |\nabla \psi| \quad (3)$$

Here, A is an advection term, B is a propagation (expansion) term, and Z is a spatial modifier term for the mean curvature k. The scalar constants α , β and γ weigh the relative influence of each of the terms on the movement of the interface. Different levelset segmentation filters use some or all of these terms in their calculations. If a specific term is not desired in the filter, then the corresponding scalar constant weighting is set to zero.

Most levelset based filters require two images as input, an initial model $\Psi(x, t = 0)$, and a feature image, which could be either the image to be segmented or some preprocessed version. The initial model should comprise of iso-value that represents surface Γ . The single image output obtained from a levelset filter is the function Ψ at the final time step where, the contour so obtained is representing the zero levelset of the output image.

The solution Γ is calculated to sub pixel precision. The best discrete approximation of the surface is therefore the set of grid positions closest to the zero-crossings in the image, as shown in Figure 5 and Figure 6.

2.2.3 Model preprocessing

No matter how good a segmented geometry is, it is generally unfit for direct CFD simulations; the reason being their indistinct or polygonal boundaries (Luca Antiga and Steinman 2006). Generally an entire segmented body is represented as a single face making it impossible to distinguish the faces comprising the inlets and outlets. In order to successfully apply boundary conditions, the boundaries carving the faces should not only be distinct, they should also be flat or planar and smooth. In addition to these requirements, the boundaries should also be circular in case Poiseuille or Womersley velocity profiles are being prescribed. Hence, the limits of the vessels should be “clipped” in a plane to define the boundaries forming the inlet and outlet faces. However, one can overcome this requirement of clipping if while segmentation the region of interest of vessel is selected such that the extreme ends of the region of interest cut the bounding box. This is the approach currently taken in our segmentation algorithm.

It is desired to extend vessels at their extremes by a length of at least 5 diameters in the direction normal to the flow so that the flow results would be devoid of any end effects. Several approaches may be taken when creating such extensions. One of the popular approaches is that of calculating the centerline of the vessel and the direction normal to the boundary at that point and extruding the inlet/outlet boundary in that direction (Luca Antiga and Steinman 2006). Another popular and simple approach is to continuously copy the boundary pixels until the desired length of extension is obtained by dilation. If possible special consideration should be given to the shape of the flow extension. Like previously mentioned, it is generally aspired to have circular cross sections at the inlet/outlets of the vessels for the purpose of applying different boundary conditions. However, the boundaries of the patient-specific vessels are never perfectly circular. Hence, in course of extending, it is desired to morph the non-circular boundary of the flow domain into a perfectly circular shape at the end of the extension. Researchers have used plane spline algorithm (Luca Antiga et al. 2008) and different morphing algorithms where the transition images are created by using simple interpolations or shapes of the levelsets of the input images (Whitaker 2000; Mukherjee and Ray 2012) for obtaining such smooth transition. After the segmented models

are processed to achieve these requirements they are deemed fit for the next step of mesh generation.

2.2.4 Mesh generation

After segmenting out the structure of the flow domain, the next step towards solving flow past a solid body is that of grid generation which represents the flow domain conforming to the segmentation. There are two major paths towards the grid generation: body conforming mesh, and non-body conforming mesh. Let us consider a case of flow past a solid cylinder in two dimensions as shown in Figure 4. Let Ω_b be the region occupied by the solid body defined by surface Γ_b while Ω_f be the region of flow domain. Where, L is the characteristic length of the body and δ is boundary layer at certain Reynolds number.

Conventionally, researchers employ structured or unstructured grids to create a body conforming mesh. It mainly involves two steps. Firstly, a surface Γ_b representing the boundary of the object is created and then a volumetric mesh is realized by considering the surface mesh as a boundary condition. So in Figure 4 a, the domain represented by Ω_f is meshed with either structured or unstructured grid. Such grid generation is often an arduous task requiring ample amount of user input. A good quality mesh needs to perfectly conform to the boundaries, be watertight, and should be devoid of any geometric singularities and small angles. This task only gets tougher as the geometries become more complicated. To tackle such complications, geometries are divided into subdomains and they are meshed individually. The common boundaries between the two domains are treated as interfaces so that the solver recognizes the domains as the region of fluid flow. Presence of multiple subdomains does add complication to the solver and so does the smoothness quotient in the interfaces. Unstructured grid is always a better choice when handling complicated geometries, however even there user needs to compromise between refinement and quality of mesh due to restrictions of singularities, small angles and sharp corners. However, once the volumetric mesh is constructed, applying boundary condition is a straight forward task especially because the faces are already defined. Various commercial software like

GAMBIT, Gridgen, Pointwise, ICEM-CFD, ANSYS Workbench etc. are available in the market which enable users to generate body fitted mesh. Afore mentioned packages are popular choices for regular geometries where object boundaries can be defined analytically or piecewise analytically in traditional engineering practices. However, in patient-specific modeling of biological systems geometries/surfaces conforming to analytical descriptions do not exist. Thus, the main challenge in patient-specific modeling is building a good surface mesh. VMTK has emerged as a popular choice when it comes to creating body fitted mesh for patient-specific vascular geometries. The method used to solve the flow will then be based on the use of structured or unstructured grids. While finite difference and finite volume approaches can be employed for structured grids, for unstructured ones finite volume and finite element methods are found suitable. Finite difference method requires governing equations to be transformed to curvilinear coordinate system which are aligned with the gridlines. This allows for the discretization of transformed equations onto the computational domain.

An alternate approach to this method is to generate non-body conforming mesh where, a Cartesian domain containing the solid body is generated irrespective of the shape of the flow domain. In this approach too, the solid body is represented by some kind of surface mesh. Let us again consider flow past a solid cylinder as shown in Figure 4 b, where, the solid boundary Γ_b actually cuts through the Cartesian volumetric grid represented by Ω_v such that: $\Omega_v = \Omega_b + \Omega_f$, where Ω_b and Ω_f are the regions occupied by solid body and fluid flow respectively. This method was first devised by Peskin in 1972 to simulate cardiac mechanics and blood flow associated with it (Rajat Mittal and Iaccarino 2005; Peskin 1972). This method completely removes the burden of generating body fitted volumetric mesh. It only requires a straight-forward task of generating an evenly spaced volumetric Cartesian grid. No matter how complicated the geometry is there occurs no change in volumetric grid, only its size and grid spacing varies with case and this is the most lucrative aspect of non-body conforming method. Since the volumetric mesh does not conform to body, mesh exists in the regions both inside and outside of the flow domain as seen in Figure 4 b. Though the flow equations get discretized everywhere, this does not account for huge

computational cost as equations only gets solved in the regions where flow occurs. Grid generation is a piece of cake for non-body conforming meshes; however the application of boundary conditions is not. Because the mesh does not conform to the boundary of the body, the equations need to be modified in the vicinity of boundary. It can be done by using finite difference, finite volume or finite element approaches without coordinate transformation. However, the cost of applying boundary condition on accuracy of the schemes and conservation properties opens a whole new can of worms.

When compared, body fitted mesh offer better control over grid resolution as the mesh contours are aligned along the object boundary. It enables having finer grid spacing in the direction normal to the boundary and account for boundary effects while retaining coarser grid spacing in the tangential direction. This leads to the requirement of fewer grid points in the flow domain opposed to evenly spaced Cartesian grid as shown in Figure 6 where d_t and d_n are the grid spacing in tangential and normal directions for a body conforming mesh while d_s is the uniform grid spacing for non-body conforming mesh. The number of grid points is found to increase with the increase in Reynolds number (Re) for non-body conforming grids. But this does not necessarily increase the computational cost of the later because the equation only gets solved in the flow domain. Further, computational cost can also be lowered by incorporating mesh refinement in the regions near boundary and trimming the extraneous mesh region that is devoid of fluid flow. On the other hand, if grid transformation is required for body conforming mesh, per grid point operation becomes higher as there are additional terms for transformation that need to be solved.

Though both the methods have their own sets of advantages and disadvantages, the major lacking of body conforming approach is observed when modeling for moving boundaries. It requires meshing a new grid at each time step and a robust procedure of projecting the solution onto a new grid [Tezduyar, 2001]. With the increase in length and frequency of motion, the procedure gets more cumbersome and erroneous. Since our lab is moving towards problems related to computing flow in moving boundaries, immersed boundary methods have been opted for grid generation which tackles such cases with ease by using stationary, non-deforming grids. Further,

when it comes to computing flow solutions in patient-specific geometries, manual labor required to set up the body fitted mesh is arduous and manipulating it is even more arduous task. So despite the progresses made in body fitted mesh IBM still remains the pick for this study because of its inherent simplicity and the nature of the problem being solved.

2.2.5 Assigning boundary conditions and computing flow

Once the volumetric mesh is prepared, the next step is to apply appropriate boundary conditions to solve the system of partial differential equations. Deciding the appropriate boundary condition in itself is massive task and a very critical one too. Vascular geometries are known to affect the local hemodynamics but boundary conditions affect large scale feature of flow like flow ratio at the site of branching and pressure distribution (Grinberg and Karniadakis 2008). Hence, physically accurate boundary conditions are mandatory for having confidence in flow results. Hence whenever possible, it is better to apply measured patient-specific velocities and pressures at the boundaries. However, it is often not possible because of invasive procedures involved in extracting those values. Researchers in such cases make wise assumptions based on physical conditions to employ boundary conditions. Describing the bases for such assumptions and boundary condition choices is out of scope of this study. However, with the recent developments in imaging modality, better and non-invasive methods for acquiring such patient-specific data have become available such as Phase Contrast Magnetic Resonance Imaging (PC-MRI), Doppler Ultrasound etc. While PC-MRI makes use of transient changes in MR field which in turn changes the phase of free hydrogen proton in blood while the phase of hydrogen atom in tissue remains the fixed. The difference in phase helps to determine the velocity of blood. Doppler Ultrasound on the other hand makes use of what is called the Doppler shift, i.e., the change in frequency of ultrasound signal when it gets reflected back from a moving target. The frequency is positive for targets that are moving towards and negative for the targets that are moving away. Both of these are non-invasive and safe and are hence popularly used in determining patient-specific values (Gefen 2012).

Be it in blood vessels, trachea, lungs, bladder or any other part of body, the main motivation behind image-based modeling is to understand disease related fluid flow. After assigning appropriate boundary conditions in fluid and solid domains the next step is to solve motion of the fluids governed by Navier Stokes equation which expresses Newton's law of conservation of momentum in fluid as it passes through a control volume in combination with equations of conservation of mass, popularly known as the continuity equation (Gefen 2012). In order to solve the flow, volumetric mesh representing the fluid domain is generally imported to in-house flow solvers or state of art commercial flow solvers like FLUENT v14.5.0 (ANSYS Inc., Canonsburg, PA), ADINA, Open FOAM etc. where flow calculations are carried out after prescribing appropriate boundary conditions. The most critical aspect about working with several commercial softwares is always the compatibility issue between files. Thankfully, most software dedicated for either image processing or mesh generation or flow simulation support various types of import and export file formats which makes this long process slightly less tedious.

For a more complete perspective on patient-specific modeling, it is required to take into consideration the happenings in solid or tissue domain but such study is out of scope of current work.

2.3 Current resources for patient-specific modeling

As mentioned in previous sections and shown in Figure 2, image based patient-specific modeling involves several steps from acquisition of image to segmentation to grid generation to assigning appropriate boundary conditions and finally computing flow. Several open as well as commercial softwares are available in market dedicated for these individual steps. For instance several packages dedicated to image processing like ImageJ, MATLAB, ITK, VTK, Slicer 3d, VMTK, Amira (Stalling, Westerhoff, and Hege 2005), Mimics ("Mimics" 2013) etc. are available that allow high quality image segmentation. Similarly, plenty of commercial software like Gambit, Gridgen, Pointwise, ICEM CFD etc., dedicated to mesh generation are also available which can

then be solved in software build for CFD analysis like FLUENT V14.5.0 (ANSYS INC., CANONSBURG, PA), Femap, Acusolve etc. to name a few.

Most researchers either use their own codes or use such software or both in combination to model and analyze patient-specific flow. Working with numerous software poses challenges of compatibility of files and efficient integrating of the workflow and hence becomes time consuming and cumbersome. Because of these reasons few researchers opt for developing their own codes. Having individual codes give the researchers liberty of modifying their codes as they find suitable and cures compatibility issues but in the meantime it also brings the possibility of errors and hence adds uncertainty constraint. Since early 2000s with the advent of ITK an idea of open-source library of image processing evolved under the initiation of National Institute of Health (NIH). Open source community has time and again proved its metal as a robust means for widespread adoption, standardization of methods, quality control, and reproducibility. With the increasing number of users and contributors from around the world, this openly shared library too emerged as an answer to the pressing need of standardization of segmentation algorithms, the constant need of debugging, and eventually extending the library. Like any other field of science it is equally crucial for the researches in the field of patient-specific modeling to be able to compare their studies and results with other results and be able to reproduce their own and others results. Actually, it is even more crucial in the field of image based modeling because these results and findings could hold clinical significance. ITK has opened doors for such possibilities and many more. VTK is also a an open source object oriented toolkit that came to realization around 2000s with the need of robust visualization tool which provides few low level image processing tools and numerous surface processing and meshing algorithms. With the advent of these open source toolkits, number of researches in the field of patient-specific modeling have increased. ITK and VTK provided robust algorithms for image analysis and visualization respectively but lacked user interface and coordination between different types of visualization (Wolf et al. 2005) which are very important for medical image analysis in clinical level. So the world of medical image analysis not only recognized the power of these toolkits but also realized how they could improve their

functionality. Hence, various researchers started devising newer toolkits that were built on the top of these two or either one of them by adding to and extending their capabilities such that it would better suit their needs.

Applications like Slicer 3D, Amira etc. and toolkits like VMTK, Simple ITK (SITK), Medical Imaging Interaction Toolkit (MITK), and Snake Automatic Partitioning (SNAP) etc. are examples of some packages that have been built on the top of ITK and VTK as their extensions, offering their own bit of advantage over their predecessor. While Slicer 3D and Amira brought several segmentation algorithms together along with 4D visualization and Graphical User Interface (GUI), VMTK focused on the need of segmenting vascular geometries particularly the ones with aneurysm and SNAP focused on extracting the homogeneous regions (Talbert et al. 2002). SITK in the mean while concentrated on making coding more simple and intuitive for those who found templates intimidating. MITK on the other hand felt the need to address the importance of 3d+t visualization for tailoring it for physicians for analyzing medical images. These are just few instances of how different groups have utilized VTK and ITK and extended it to fit their needs.

2.4 Our approach for image-based modeling

Our thermo fluid group is also well aware of the power and robustness of ITK and VTK and their contribution towards the patient-specific modeling society, especially this being one of our major research interests. Since our aim is to study fluid dynamics in a range of physiological conditions related to blood vessels to valves to larynx to bladder etc., our lab deals with different imaging modalities too. Further, our focus is in bypassing the mesh generation steps involved in conventional patient-specific modeling and opt for a route involving levelset based segmentation to solve flow (See Figure 7), which has been proved to generate better flow results (Dillard et al. 2014). Though ITK and VTK are oceans of segmentation and visualization codes and meet lots of our segmentation needs, they still fall short for our purpose due to the lack of GUI and few capabilities that are required for our research. As discussed earlier, though their surrogates are available, none of them completely meet our needs.

Furthermore, our lab already has a robust setup for segmenting medical images which is based on levelset methods and bypasses the tedious path of mesh generation (Dillard et al. 2014) as shown in Figure 3 and Figure 7. However, as indicated in Figure 7 and Figure 8, our current route involves going back-and-forth from one code to another and one software to another. For instance, even obtaining a small region of interest involves downsampling of original DICOM image in MATLAB and opening it in some image viewer like Tecplot and locating the region of interest which is then followed by cropping in MATLAB. So, it was high time for our group to bring the capabilities of numerous codes and software together in one platform to make the methodology straight forward, organized, efficient and less cumbersome. Also from an end users' perspective, it is a valuable addition to have a GUI that binds all the algorithms together. This led to dreaming about creating our own toolkit with a user interactive GUI on the top of ITK and VTK not only giving access to all of their capabilities but also allowing additions of our own.

As discussed earlier, we aimed to steer away from the conventional route of image-based modeling methods that rely on body-fitted mesh because of the pain involved in performing FSI problems. Instead, a route has been chosen which is based on Cartesian mesh and which uses levelsets to define embedded boundaries. However, though our approach is robust, it is still tedious and cumbersome because of numerous software and codes involved. Hence, this is an attempt to address the pitfalls by bringing all the steps of image-based modeling setup and CFD into one unifying platform; which generates embedded boundaries representing the object of interest obtained from levelset based segmentation by allowing a user interactive GUI and aiding visualization. In this work, geodesic active contour approach is employed for image segmentation to obtain levelset field comprising of boundaries of object of interest which is then be fed into parallel Eulerian Lagrangian Algorithm for Interfacial Transport (pELAFINT) solver to compute flow. If needed, surface representations of the embedded boundary can also be created using marching cubes and the model can be manipulated or processed as deemed necessary. And not to forget these entire operations can now be performed in one platform which is equipped with GUI.

CHAPTER 3. BUILDING A USER INTERACTIVE TOOLKIT

To achieve our goals listed in section 1.2, a setup is needed where all the capabilities of image-based modeling can be brought together and can be equipped with GUI such that a user interactive toolkit for image-based modeling can be obtained. Our search ended with IA-FEMesh toolkit, a toolkit especially designed for musculoskeletal modeling using finite element methods by Musculoskeletal Imaging, Modeling, and experimentation (MIMX) group of The University of Iowa. Some of the frame work required for our purposes like GUI already existed in forms that were specialized for musculoskeletal modeling purposes within IA-FEMesh toolkit. Further, it was built on the top of ITK and VTK libraries which are standard state of the art libraries widely used for segmentation and visualization purposes as indicated in 2.3 and 2.4. Hence, we decided to collaborate with MIMX group and borrow the setup of IA-FEMesh so that the capabilities specific to cardiovascular and bio fluid modeling for CFD could be developed within it. The basic software framework of our toolkit and the capabilities that already existed and that have been added are discussed in this chapter.

3.1 Framework of IA-FEMesh toolkit

IA-FEMesh is a project of MIMX Program (Grosland et al. 2009). It is a collaborative effort directed at computational modeling of anatomic structures. A primary objective is to automate the development of patient/subject-specific models using a combination of imaging and modeling techniques, with particular emphasis on finite element (FE) modeling. A primary goal of MIMX is to develop a software package (IA-FEMesh) composed of orthopedic-related FE model development capabilities that will ultimately allow surgeons and engineers to work together to plan patient-specific care using objective analysis-based tools. Their objective is to better evaluate alternate surgical protocols and treatments, thereby contributing to improved surgical outcome.

IA-FEMesh is wrapper software written in C++ and built on the top of ITK, VTK and KWWidget to facilitate robust and interactive anatomic Finite Element (FE) modeling

experience. It is freely available and offers a range of capabilities built towards generating a hexahedral FE mesh. It also enables visualization and supplies tools for mesh quality evaluation. The development team has put special effort towards making FE modeling user friendly and interactive experience.

Though IA-FEMesh has attained most of its goals as far as FE modeling is concerned, it only offers limited facilities for images which include loading and deleting image and basic segmentation capabilities apart from their visualization and of a surface/mesh overlaying the image. Since IA-FEMesh has the framework required for image processing for the purpose of image segmentation, our thermo fluid group thought it would be worthwhile to develop much needed user interactive segmentation capability for our group using the IA-FEMesh framework.

3.1.1 Visualization tool kit (VTK)

VTK is an open source, freely available tool kit that offers 3d interaction techniques with data. It offers 3D computer graphics, image processing and visualization. It consists of a C++ class library and several interpreted interface layers including Tcl/Tk, Java, and Python. Kitware, whose team created and continues to extend the toolkit, offers professional support and consulting services for VTK. VTK supports a wide variety of visualization algorithms including: scalar, vector, tensor, texture, and volumetric methods; and advanced modeling techniques such as: implicit modeling, polygon reduction, smoothing, cutting, contouring of meshes, and Delaunay triangulation. VTK has an extensive information visualization framework, has a suite of 3D interaction widgets, supports parallel processing, and integrates with various databases on GUI toolkits such as Qt and Tk. VTK is cross-platform and runs on Linux, Windows, Mac and Unix platforms.

3.1.2 Insight segmentation and registration toolkit (ITK)

ITK is an open-source, cross-platform system that provides developers with an extensive suite of software tools for image analysis. ITK employs leading-edge algorithms for registering and segmenting multidimensional data. ITK provides numerous functionalities of image

processing. It is an image processing engine but it does not provide any graphical interface or visualization facilities. There are many GUI toolkits available, many of which are cross-platform and can be used as an extension of ITK for visualization purposes. A common build environment called CMake is used to manage the compilation process. ITK is specifically developed for segmentation and registration of medical images by a team with expertise in a wide range of areas related with registration and segmentation.

ITK is implemented in C++ programming language. It uses generic programming style of C++ that extensively uses templated code which makes it highly efficient. This feature has also led to easy discovery of many software problems at compile-time, rather than at runtime. ITK uses a model of software development called as extreme programming, which is a simultaneous and iterative process of design-implement-test-release which emphasizes on customer satisfaction and considers managers, programmers and customers as equal partners (“Extreme Programming” 2013). This approach helps to manage the rapid evolution of the software. ITK is also tested every day using Dart. Dart is an open-source, distributed, software quality system that allows software projects to be tested at multiple sites in multiple configurations (hardware, operating systems, compilers, etc.).

Some of the advantages of ITK:

Since ITK provides automated wrapping process, it is possible to develop and customize the application using variety of programming languages such as TCL, Java, and Python. ITK is organized around data-flow architecture: Data is represented using data objects which are in turn processed by process objects (filters). Data objects and process objects are connected together into *pipelines*, a chain of operations dedicated for getting a result. This loosely coupled architecture allows efficient code extension and re-usability as well as variety of data formats. Support for our own data format can be added to develop the application using ITK. ITK can process images of N-dimension. ITK facilitates efficient memory management through the use of *smart-pointers* by using garbage collection in an efficient manner. That is, “smart pointers” can be allocated on the stack, and when scope is exited, the smart pointers disappear and decrement their reference count

to the object that they refer to. ITK supports multi-threading which allow developing programs that run on parallel processors to facilitate data parallelism.

3.1.3 KWWidgets

KWWidgets is a free, cross-platform and open-license GUI Toolkit. It provides low-level core widgets, advanced composite widgets, and high-level visualization-oriented widgets that can be interfaced to visualization libraries like VTK. KWWidgets is an object-oriented C++ toolkit that can interact and co-exist with Tcl/Tk directly from C++. It is wrapped automatically into a Tcl package, and therefore can be used directly from Tcl/Tk also, allowing for fast-prototyping and scripting.

KWWidgets can be compiled on a large number of platforms, including Windows, Mac, and many flavors of Unix/Linux systems. It is open-source and uses the same BSD-style open-license as VTK. It is available for free and can be downloaded without restrictions. Like many other GUI toolkits, it provides low-level core widgets like buttons, entries, scales, menus, combo-boxes, thumbwheels, spin-boxes, trees, notebooks and multi-column lists to name a few. Unlike many of those toolkits though, it also provides advanced composite widgets like toolbars, tooltips, progress gauges, split-frames, splash-screens, 2D/3D extents, color pickers, histograms, windows and dialogs.

3.1.4 CMake

CMake is an open-source software system that is used to compile and build softwares in an environment that is platform (UNIX, Windows) and compiler (intel, gnu, gcc etc.) independent. It produces native build files appropriate to these Operating systems. It creates makefiles for Unix and generates projects and workspaces for Windows. Basically, CMake generates a build environment that generates make files, wrappers, compiles source codes, creates libraries and builds executable. It supports both in place build, in the same source folder, as well as out of place build that is in any folder other than the source directory. It was originally developed by Kitware, as part of the ITK project.

3.2 Targeted capabilities of IA-FEMesh:

It has been more than a decade that image based modeling has come into practice (Mittal and Iaccarino 2005) but all the research groups either using body fitted mesh or non-body conforming mesh methods have found it to be a struggle to bring all the capabilities required in one platform. Our main target in this thesis is to gather all of those capabilities together in one platform that has specialized levelset based segmentation features well suited for image based modeling of biological systems such as blood vessels, bladder, trachea etc. along with a user interactive and user friendly GUI. For this purpose, efforts have been put towards bringing together the following capabilities in IA-FEMesh such that all the steps from image segmentation to preprocessing for solver can be performed in one platform of IA-FEMesh.

1. Image segmentation
2. Manipulation of segmented object in image domain
3. Preparation of model for flow solver

Image segmentation is a process of outlining the boundaries of different entities present in an image and distinguishing them from one another as our eyes would perceive them. Segmentation can be carried out based on pixel intensity or gradient or minimization of functional but for the purpose of accurately delineating real images one requires a mixture of these methods. Levelset based segmentation methods are incorporated in this toolkit which treats an image and its pixel values as a topological field of sorts, on which segmentation can be accomplished by computing pixel gradients or by minimizing a functional over the pixel field in a manner analogous to energy minimization. Read more on this in Chapter 4.

Though lots of commercial software are capable of modifying regular geometries and meshes, modification of patient-specific geometry is an extremely difficult task. Most researchers stay away from image-based modeling because of this very obstacle as it limits the scope of research. IA-FEMesh is equipped with several tools for manipulating musculoskeletal objects, however most of them need to be tuned for cardiovascular and other bio fluids set up and the

requirement of image based modeling varies from that of musculoskeletal modeling. Read more on this in Chapter 6.

Similarly, set up for assigning boundary conditions for skeletal operations too already exists in IA-FEMesh. However, the pre-existing framework for the same is significantly different than the one required for CFD. Hence efforts were put toward tuning these capabilities and modifying them to suit our needs. Read more on this in Chapter 5.

CHAPTER 4. IMAGE SEGMENTATION CAPABILITY

The first capability aimed and developed in this interactive toolkit is that of segmenting images from different modalities and generating a levelset representation of the same which can then be used as an input to our in-house flow solver – parallel Eularian Lagrangian Algorithm for Interfacial Transport (pELAFINT). The accuracy of the segmentation algorithm of the toolkit was measured by segmenting a CTA image of phantom of Intra Cranial Aneurysm (ICA) and comparing the dimensions with the phantom itself. For a successful segmentation of any image, simple or sophisticated, it is crucial that user understands the approach undertaken, significance of individual parameters of filters and implication of their variance. Hence, users are provided with guidance regarding each step and parameters involved in segmentation. To ensure that users have enough guidance and the functionalities have been arranged in an intuitive manner to successfully segment images, a *usability test* was also carried out.

4.1 Image analysis

Digital images are composed of pixels/voxels that possess certain intensity or color value. The range of pixel values in an image may vary according to the image type. Generally pixel intensities of a gray scale image range from zero to 255 where zero represents black and 255 represents white. Color images follow similar pattern but red, green and blue vector patterns replacing scalar intensity levels. E.g. black has a vector representation of (0,0,0), while medium green has a representation of (0,125,0), etc. A color image has 256x256x256, or about 16.8 million, possible pixel hues. Groups of pixels give an image its meaning, concertedly transforming it from an array of numbers or vectors to a picture that can be visually related to something real. More pixels describing a scene yield finer detail, or better resolution, and lend greater fidelity to objects being depicted.

4.2 Objective 1. Build efficient image segmentation capability

Modern segmentation methods in the literature treat an image and its pixel values as a topological field of sorts, on which segmentation can be accomplished by computing pixel gradients or by minimizing a functional over the pixel field in a manner analogous to energy minimization.

In the current image-to-computation framework, levelset representations are unifying elements of the segmentation and Cartesian grid flow solver algorithms. The main objective behind image segmentation is obtaining a levelset field that will be used in applying boundary conditions during fluid flow calculations. Our Thermo fluids Group uses levelset based Cartesian grid method to analyze flow in patient-specific geometries. This method is capable of solving moving boundary problems too.

ITK provides a basic set of algorithms that can be used to segment images of different modalities. These algorithms can also be used in combination to obtain better quality segmentation application. In fact, though numerous methods of segmentation are available today, no single method works for all kinds of images. The most effective segmentation can be obtained by customizing combinations of the basic segmentation algorithms and pre-processing and post-processing algorithms implemented in ITK for images of specific modality. Depending on the modality of the input image and the anatomical region of interest, the parameters of the filters need to be fine-tuned for better segmentation.

Following are the filters we have implemented for general image segmentation: CropImageFilter, RescaleIntensityImageFilter, SigmoidImageFilter, ThresholdImageFilter, FastmarchingImageFilter, SignedMaurerDistanceImageFilter, BinaryThresholdImageFilter and GeodesicActiveContourImageFilter. These filters are discussed in detail in section 4.3.

4.2.1 The levelset methods for image segmentation

Levelset method, one of the forms of deformable models, was first introduced by Osher and Sethian for front propagation but it was later used by Malladi for image segmentation. The main crux behind active contours (deformable models) is that a user has to specify an initial guess for the contour. This initial contour then gets propagated by image driven forces to the boundaries of the desired objects. In such models, two types of forces are considered - the internal forces, defined within the curve, are designed to keep the model smooth during the deformation process, while the external forces, which are computed from the underlying image data, are defined to move the model toward an object boundary or other desired features within the image.

There are two forms of deformable models: explicit and implicit (Angelini, Jin, and Laine 2004). In the explicit deformable model, also known as parametric form, (also referred to as snakes), an explicit parametric representation of the curve is used. In contrast, the implicit deformable models, (also called implicit active contours or levelsets), are designed to handle topological changes naturally. In a levelset approach, instead of following the interface/contour directly, the contour is built into a surface. That is, contour is embedded as the zero levelset of a higher dimensional function called the levelset function (Chen Francis). The levelset function is then evolved under the control of a partial differential equation. The motion of the interface is matched with the zero levelset of the levelset function, and the resulting initial value partial differential equation for the evolution of the levelset function resembles a Hamilton-Jacobi equation. In this setting, curvatures and normals may be easily evaluated, topological changes occur in a natural manner, and the technique extends trivially to 3 dimensions. At any time, the evolving contour can be obtained by extracting the zero levelset from the output.

Fast marching levelset segmentation, active contours, active contour without edges (Chan and Vese 2001), threshold levelset segmentation (A. Xu et al. 2010), shape detection levelset segmentation (Malladi, Sethian, and Vemuri 1995), geodesic levelset segmentation (Caselles, Kimmel, and Sapiro 1995) are few of the popular segmentation approaches based on levelset methods where each methods have their own set of advantages and limitations.

4.2.2 Levelset based two step segmentation to define embedded boundaries

Accuracy is a crucial element in image based modeling as the results could have clinical implications. No matter how robust any flow solver is, it cannot give reliable flow results unless the surface boundary fed into it accurately represents the physical entity. So the choice of segmentation algorithm directly affects the outcome of a CFD simulation. Hence our goal was to seek for a levelset based segmentation algorithm that could capture the modeled geometry accurately and smoothly. The segmentation approach adopted in our framework is largely based on two step levelset based segmentation approach devised by Hernandez et al. in 2003 (Juan R Cebra et al. 2005; Hernandez et al. 2003), where fast marching algorithm is used to obtain a fast and crude segmentation of the ROI which is then followed by geodesic active contour based segmentation to achieve final smooth geometry.

The fast marching method introduced by Sethian as an efficient approach for solving Eikonal equation. It takes a stationary approach to solving moving boundary problems. Consider a curve Γ_0 in two dimensions evolving along its normal direction with a positive speed F . Let T be the time required by the curve to reach point p . Then the function T represents the levelset of the evolution of the curve Γ_0 . This function T is the solution of the Eikonal equation

$$F \cdot |\nabla T| = 1 \quad (4)$$

With initial condition $T(\Gamma_0) = 0$. The evolution of Γ_0 at time t is the levelset t of T . Operator has to select a point inside the region of interest for initializing the algorithm. Front propagates from the selected point with the speed calculated from the feature image and an *action map*, record of arrival time, is obtained as a result. Such action map enables operator to obtain snap shot of propagating front at any time and allows selection of appropriate levelset t giving rough segmentation of the ROI.

This crude segmentation is followed by a segmentation based on geodesic active contour method for obtaining smoother segmentation. The geodesic active contour technique put forward

by Caselles, Kimmel, and Sapiro is based on the relation between active contours and the computation of minimal distance curves also known as geodesics which lie on Riemannian space. This approach brings together classical “snakes” based on energy minimization and geometric active contours based on the theory of curve evolution. The crux of the classical “snake” approach is deforming an initial contour C_0 towards the boundary of the region of interest. Such deformation is obtained by minimizing a functional designed so that its (local) minimum is obtained at the boundary of the region of interest. Geometric models (Levelset approaches) on the other hand, are based on the theory of curve evolution and geometric flows (PDE) based on mean curvature motion. In these type of active contours models, the curve is propagating (deforming) by means of a velocity that contains two terms as well, one related to the regularity of the curve (curvature term) controls the smoothness of the curve and the other (propagation term) shrinks or expands it towards the boundary.

While implicit active contour models avoid several of the difficulties known from explicit models such as inability of adapt to topological transformations (T. McInerney and Terzopoulos 1995), the notable advantages offered by this formulation include: no parameterization of contour, splitting and merging of objects, good numerical stability, and straight forward extension of the 2D formulation to n-D (Angelini, Jin, and Laine 2004). Their main disadvantage is additional computational complexity. First, in their simplest implementation, the partial differential equation must be evaluated on the complete image domain. Second, most approaches are based on explicit updating schemes which demand very small time steps.

Geodesic active contour improvises previous models of geometric (PDE) active contours allowing stable detection of boundaries when the gradients defining boundaries have high fluctuation in intensity and even gaps. Levelset representation of Geodesic active contour model as derived from energy functional is as under.

$$\frac{\partial u}{\partial t} = |\nabla u| \operatorname{div} \left(g(I) \frac{\nabla u}{|\nabla u|} \right) + cg(I) |\nabla u| \quad (5)$$

The solution to the object detection problem is then given by the zero levelset of the steady state ($u_t = 0$) of this flow. As described in the experimental results, it is possible to choose $c = 0$ (no constant velocity), and the model still converges (in a slower motion). This model has less parameter (with $C = 0$) as compared to levelset equation.

Above geodesic flow includes a new component in the curve velocity that improves those models. The new velocity component allows us to accurately track boundaries with high variation in their gradient, including small gaps, a task that was difficult to accomplish with the previous curve evolution models.

4.2.3 Deriving levelset information to supply to the flow solver

This section represents the crux of the image-to-model portion of our framework, and gives an outline of the complete set of steps involved with converting imaged data to modeled surfaces for use in flow simulations. The entire procedure, as illustrated in Figure 9, is demonstrated, starting with a CTA image, then through pre-processing, segmentation and ending with a 3D reconstruction of an intracranial aneurysm.

Our first step is to crop the original image so as to isolate the ROI as much as possible (Figure 9(a)), so that one can get rid of many features that are redundant or are out of our ROI in the original image. This also improves efficiency and reduces computational time in cases like in Figure 9 of ICA geometry, which only occupies a small part of the complete CTA image. As the second step, the image was rescaled to the intensity of 0.0 to 1.0 and was prepared for sigmoid filter where the desired regions of the image was highlighted by selecting four distinct parameters. Once the regions of interest have been enhanced, many redundant features appearing in the resultant image can be thresholded out to obtain a final edge image. Such edge image is introduced to fast marching filter and then the users are allowed to place initial guess in the form of seeds of certain radius. These seeds are initial levelsets which are allowed to evolve in a specific direction based on the speed image for certain iteration so that initial guess image is obtained. Signed distance of this image is calculated using Signed Maurer distance function so as to produce a

levelset representation of the guess image that can be supplied to the geodesic active contour filter. Edge image obtained from sigmoid filter is also input to this filter as feature image. Users are allowed to tune the parameters relating to advection, propagation and curvature and number of iterations. Higher iterations could lead to leaking of the contour outside of the physical boundary given by edge image.

Once the desired object has been isolated and extracted from the image domain, its levelset field is mapped to a flow solver mesh with local mesh refinement and pruning capabilities (described in the next section) for CFD simulation.

Only the capabilities incorporated in IA-FEMesh for segmentation will be discussed in this chapter. The other two capabilities will be discussed in the following chapters.

4.3 Pipeline for general segmentation (Method I)

As discussed in section 4.2.2, geodesic active contour based segmentation is a robust method that gives smooth images and conserves local curvatures of the flow domain which are important factors for computing flow equations. Hence, this method has been chosen for segmentation and a pipeline connecting several filters have been set up which would enable carrying out segmentation based on geodesic active contours. Another important factor contributing to the selection was that it was based on levelset method and would give us a levelset field as final output which is the required form of input image for our flow solver.

In the pipeline shown in Figure 10, CropImageFilter, RescaleIntensityImageFilter, SigmoidImageFilter, and ThresholdImageFilter are the preprocessing filters supplying input images in the format required for levelset based segmentation achieved by utilizing FastMarchingImageFilter and GeodesicActiveContourLevelSetImageFilter. This pipeline will be termed as Method I in this work.

This pipeline works by constructing a “feature image” onto which the evolving levelset locks as it moves. In the feature image, values that are close to zero are associated with region outside of the object while values close to 1.0 are associated with the inside of the object. An

original (or preprocessed) image is given to the filter as the feature image. When an original image contains good contrast and the object of interest has fairly high intensity than its background then filters used for preprocessing purposes like `SigmoidImageFilter` and/or `ThresholdImageFilter` need not be used. A seed(s) for the levelset is given as the input of the `FastMarchingImageFilter`. The seed is converted into a levelset embedding which propagates according to the features calculated from the original/preprocessed image.

A contour is represented in the form of a zero levelset of a function. The function evolves and carries with it the zero levelset. Evolution of the function is controlled by a partial differential equation in which a speed term is involved. This speed is computed based on the feature image provided by the user. The differential equation includes an additional term representing an inflation force. The strength of this force can be regulated in order to make the contour expand and eventually overcome small edges. In order to initialize the `GeodesicActiveContourLevelSetImageFilter`, an input levelset is required. It is desirable for this initial contour to be relatively close to the final edges of the anatomical structure to be segmented. In this application, a `FastMarchingImageFilter` is used to produce a rough contour from a set of seeds. In order to initialize the `FastMarchingImageFilter`, the user selects a set of seed points in the viewer of the input image. The fast marching filter will propagate a front starting from the seeds and traveling at constant speed until the front reaches a user-specified distance. At that point the front is passed to the `GeodesicActiveContourLevelSetImageFilter` which will use the partial differential equation in order to continue the evolution of the contour. The `GeodesicActiveContourLevelSetImageFilter` also uses a speed computed from the speed image. This should result in the contour slowing down close to object edges. The final output of the filter is a levelset in which the zero set represents the object borders. The number of iterations to run this filter is a critical parameter since too much iteration will always result in the contour leaking through the regions of low contrast of the anatomical object borders.

All of the filters in this levelset based segmentation set up operate with floating point precision to produce valid results.

4.3.1 CropImageFilter:

In order to seclude the region of interest from an otherwise large image, the portions of image that did not hold significance for our purpose were cropped out. It was possible to declare a region of interest within an image without cropping the image, but as for the requirement of our solver in order to supply boundary conditions in the Cartesian grid, the ends of the vessels needed to meet the planes of Cartesian grid domain. Cropping in such manner has also voided the need of capping the ends of the vessel to create a closed region for segmentation. A class called `itkCropImageFilter.h` was applied to remove the pixels from upper and lower bounds of the largest possible regions so that only the region of interest could be retrieved. This class made use of `itkExtractImageFilter.h` class available in ITK. Incorporating this filter in IA-FEMesh has provided our group with a better alternative than the traditional and cumbersome route involving different softwares like Matlab, Tecplot as shown in Figure 8 and discussed in sections 1.1 and 2.4.

4.3.2 RescaleIntensityImageFilter:

Images considered for flow studies can vary in their intensity ranges depending upon the modality of image acquisition. Most of the medical images undertaken for flow studies are Digital Imaging and Communications in Medicine (DICOM) image which have intensity ranging from [-2048 to +2048]. Since the goal is to build a pipeline applicable for most of image types and have less parameters that required user intervention, a conscious decision was made to tune the all the filters such that their operating intensity range would be 0.0 to 1.0. Hence, a capability of rescaling the intensity of an image was added to the pipeline. This filter requires input image to have floating point pixel values so that the intensity range does not get altered and important information do not get lost while rescaling.

An input image is hence scaled to the range of 0 to 1.0 using `itkRescaleIntensityImageFilter` before it is fed into `itkSigmoidImageFilter` for obtaining

a feature image. This filter utilizes `itkRescaleIntensityImageFilter.h`, a header file, from ITK library. This filter is responsible for performing pixelwise linear transformation of intensities of an image. It utilizes following equation to map the intensities and calculate `OutputPixel` corresponding to each `InputPixel`.

$$OutputPixel = (InputPixel - InputMin) \cdot \frac{(OutputMax - OutputMin)}{(InputMax - InputMin)} + OutputMin \quad (6)$$

Where, `InputMax` and `InputMin` are the maximum and minimum intensity values of input image and `OutputMax` and `OutputMin` are the maximum and minimum values of intensities that are desired in output image. The pixels are treated as real type in this filter.

4.3.3 SigmoidImageFilter:

`SigmoidImageFilter` is used to create feature or speed image for our segmentation pipeline. This filter can be considered as a contrast enhancement filter generally used as a preprocessing tool. Since medical images and in fact most real images do not have much contrast between the background and the ROI, it is generally difficult to create a good speed image required for `FastMarchingImageFilter`. `SigmoidImageFilter` helps overcome this difficulty to a great extent. It is used as a tool for pixel-wise intensity transformation. This filter is implemented using the class `itkSigmoidImageFilter.h` in ITK. `SigmoidImageFilter` allows a user to highlight a range of intensities of input image and smoothly attenuate intensities beyond it. If the cropped/rescaled image shown in Figure 9 (b), where the region of interest as well as noise appear to have almost same intensity (since the image was colored by intensity), is compared with the sigmoid image shown in Figure 9 (c), where the object of interest colored in light green looks distinct from the surrounding noise colored in blue and orange/brown, then it becomes easier to appreciate the contribution of sigmoid function.

If 'x' be the input pixel intensity then then the output pixel intensity 'y' obtained after Sigmoid intensity transfer is given by the following equation:

$$y = (M - m) \frac{1}{1 + e^{-\frac{x-\beta}{\alpha}}} + m \quad (7)$$

Where, M and m represent the maximum and the minimum output image pixel intensities, respectively. β is a parameter that controls the pixel value around which the intensity of interest is centered. α is a parameter that controls the range of input intensity that are in our interest. For instance, if the object of our interest has pixel intensity of 150 or above, then our β should be around 150. Now, the selection of α will indicate the range of intensity that one wants to highlight or variance that one wants to have. If the desired foreground belongs to a range of [140-160] then α should be 10. Larger value of alpha would give smoother image but larger amount of noise too. Similarly, lower value of alpha will give sharp image with negligible amount of noise. This filter treats intensities as real type.

In current pipeline, the values of maximum and minimum pixel intensities have been fixed to be 1.0 and 0.0 respectively for the reasons of consistency discussed in previous section 4.3.2. Hence, though α and β can have negative as well as positive values, in our set up their values can only be positive. The resulting image is again a real image that can be used as feature input image to `FastMarchingImageFilter` as well as `GeodesicActiveContourLevelSetImageFilter`. However, another filter for thresholding out unwanted intensities clearly observed in the output of `SigmoidImageFilter` has been added in the pipeline.

4.3.4 `ThresholdImageFilter`:

Output image resulting from `SigmoidImageFilter` is again subjected to be threshold to remove noise by using `itkThresholdImageFilter.h` class. It sets the intensities that are above, below or between simple threshold value to a user specified 'outside' value. In our pipeline, the resulting image obtained as such will be used as a feature image for both `FastMarchingImageFilter` as well as `GeodesicActiveContourLevelSetImageFilter`. This filter treats intensities as real type.

4.3.5 FastMarchingImageFilter:

This filter implements a fast marching evolution algorithm called fast marching to a simple levelset evolution problem. A contour is represented in the form of a zero levelset of a function. The function evolves and carries with it the zero levelset. Evolution of the function is controlled by a partial differential equation in which a speed term is involved. This speed is computed based on an image provided by the user. A speed image is produced by a mapping of the original image. In our set up, the mapping is created by using sigmoid function in such a way that regions with high contrast will have low speeds while homogeneous regions will have high speed. This is very convenient since the sigmoid offers four parameters that can be tuned in order to select the features of interest from the input image. As shown in the form of two red spheres in Figure 9 (d), the user selects a set of seed points in the viewer of the input image. The FastMarchingImageFilter will propagate a front starting from the seed(s) and traveling with the speed computed from the speed image. This should result in the contour slowing down close to object edges. The result of the filter is a *time-crossing map* which indicates, for each pixel, how long it took to the contour to reach at this pixel location. This map is thresholded by implementing BinaryThresholdImageFilter in order to capture a particular snapshot in time. Time thresholding is carried out in order to obtain the required front at required time. Selecting the appropriate time is very critical and it should be done such that the resulting object is as close to final image as possible and at the same the contour should not leak out of image boundary too. The results obtained after applying FastMarchingImageFilter onto image of ICA at different time steps 60, 70, 80 and 90 have been shown in Figure 11. For this case of ICA, best result was obtained at time step of 90 (also included in Figure 9 (e)), after that contour leaked outside of image boundary. It is clear from the above images that this algorithm has potential of capturing the object of interest but the geometry so obtained will be pixelated and not smooth hence not directly usable for flow computation. However, this limitation can be easily overcome by applying geodesic active contour based algorithm that follows.

4.3.6 SignedMaurerDistanceMapperFilter:

In order to feed a levelset input into geodesic active contour filter, a signed distance image is created from the thresholded output of `FastMarchingImageFilter`. This is created with the aid of `itkSignedMaurerDistanceMapperFilter.h` class which calculates the Euclidean distance transform of a binary image in linear time for arbitrary dimensions (Maurer, C.R., Qi, and Raghavan 2003).

4.3.7 GeodesicActiveContourLevelSetImageFilter:

Like `FastMarchingImageFilter`, this filter too implements a levelset approach to segmentation. A contour is represented in the form of a zero set of a function. The function evolves and carries with it the zero levelset. Evolution of the function is controlled by a partial differential equation in which a speed term is involved. This speed is computed based on an image provided by the user.

The differential equation includes an additional term representing an inflation force. The strength of this force can be regulated in order to make the contour expand and eventually overcome small edges. In order to initialize the `GeodesicActiveContourLevelSetImageFilter`, an input levelset is required. It is desirable for this initial contour to be relatively close to the final edges of the anatomical structure to be segmented. In this application, `FastMarchingImageFilter` is used to produce a rough contour from a set of seeds. In order to initialize the `FastMarchingImageFilter`, the user selects a set of seed points in the viewer of the input image. The `FastMarchingImageFilter` will propagate a front starting from the seed(s) and traveling at constant speed until the front reaches a user-specified distance. At that point the front is passed to the `GeodesicActiveContourLevelSetImageFilter` which will use the partial differential equation in order to continue the evolution of the contour. The `GeodesicActiveContourLevelSetImageFilter` also uses a speed computed from the speed image. This should result in the contour slowing down close to object edges. The final output of

the filter is a smooth levelset field in which the zero levelset represents the object borders as shown in Figure 9 (f). The number of iterations to run this filter is a critical parameter since too much iteration will always result in the contour leaking through the regions of low contrast of the anatomical object borders.

The `MaximumRMSChange` parameter is used to determine when the solution has converged. A lower value will result in a tighter-fitting solution, but will require more computations. Too low a value could put the solver into an infinite loop unless a reasonable `NumberOfIterations` parameter is set. Values should always be greater than 0.0 and less than 1.0. In our pipeline, this value has been set to 0.002. The `NumberOfIterations` parameter can be used to halt the solution after a specified number of iterations, overriding the `MaximumRMSChange` halting criteria. The `CurvatureScaling` parameter controls the magnitude of the curvature values which are calculated on the evolving *isophote*. This is important in controlling the relative effect of curvature in the calculation. Default value is 1.0. Higher values relative to the other levelset equation terms (propagation and advection) will give a smoother result. However if this value is very high then vessels with small diameters may collapse as a result. This can be seen in an exemplary results of segmentation shown in Figure 25. The `PropagationScaling` parameter controls the scaling of the scalar propagation (speed) term relative to other terms in the levelset equation. Setting this value will override any value already set by `FeatureScaling`. The `AdvectionScaling` parameter controls the scaling of the vector advection field term relative to other terms in the levelset equation. Setting this value will override any value already set by `FeatureScaling`.

In general, ITK levelset filters implements a levelset approach to the segmentation. A contour is represented in the form of a zero levelset of a function. The function evolves and carries with it the zero levelset. Evolution of the function is controlled by a partial differential equation in which a speed term is involved. This speed is computed based on an image provided by the user.

4.4 Segmentation of aorta (Method II)

Pipeline used in previous section, Method I, works well with most patient specific geometries like blood vessels, trachea, segments of brain etc. However, at times we come across images in which entities of our interest are connected with other entities that are of no significance to us. These unwanted entities may be tissues or noise of similar intensities which are attached to the boundaries of the object of interest. This is often the case while segmenting aorta (both healthy and diseased). Sections of bone in ribcage is often found attached to the aorta via connectors. And to make matters more complicated the outer region of bone often have same intensity as an aorta. As a result, the segmented result will have many extra connected features. Hence the preprocessing steps used in previous segmentation pipeline are inadequate to seclude the object of interest in such image as the chances of leakage are higher. The pipeline prepared (see Figure 22) for preprocessing cases like aorta involves the following filters:
itkConnectedThresholdImageFilter, itkErodeImageFilter,
itkConnectedThresholdImageFilter, itkDilateImageFilter,
itkConnectedThresholdImageFilter.

itkConnectedThresholdImageFilter is essentially a Region growing filter which allows user to place initial seed points within in the image (see Figure 26 (i)) and supply the upper and lower threshold limits of the intensities to be segmented. The seed is allowed to propagate only through the voxels that are connected to the location of the seed and lies within the upper limit and lower limit of the intensities supplied by the user. The result of this filter is a binary image which is then subjected to erosion by the number of voxels supplied by the user. The purpose of the erosion at this point is to remove the connectors. The resultant image is again filtered through itkConnectedThresholdImageFilter to obtain a clean segmentation of the object of interest. The resultant binary image is the dilated using itkDilateImageFilter by the same number of voxels used for erosion. An image of the resultant binary image superimposed over the original image for clarity is shown in Figure 26 (ii). This resultant binary image can be used as an input to FastMarchingImageFilter and GeodesicActiveContourLevelsetImageFilter filter to

obtain smooth levelset representation of the object of interest (see Figure 26 (iii) & (iv)). In case the object requires manipulation like extensions etc. then this binary image can be used for that purpose and the image after manipulation can be supplied for two step levelset based segmentation method. This pipeline will be referred to as Method II in this work.

4.5 Segmentation of aortic dissection (Method III)

In addition to the noise, calcifications and connectors linking aorta to close structures like bones, aortic dissection poses special complication to image segmentation due to the complicated physiology. Presence of true and generally multiple false lumens results in blood moving with different velocities at different lumens within an aorta at any chosen time. The resultant scan of the dissected aorta hence has regions of highly varying intensities (Krissian et al. 2014) which makes it difficult to rely solely on intensity based methods to obtain the object of interest. The second challenge encountered during segmentation aortic dissection is fuzzy boundaries created as a result of fast motion of the dissection wall which often appears as duplication of wall. The third challenge was due to non-homogeneous contrast of dissection wall which makes it difficult to separate a wall from a false lumen and background. The fourth major challenge encountered was the condition of collapsed true lumen causing true lumen to be only a voxel thick sheet adding limitation to the selection of filters.

In the past few years a few groups have developed algorithms for detecting and segmenting the wall of aortic dissection. In general these algorithms have focused on tracing of outer lumen of aorta followed by the wall of aortic dissection. Variations of above approach have been taken by researchers working on tracing dissection. For instance Krissian et al. (Krissian et al. 2014) used multiscale vesselness measure (a measure that indicates how similar the local structure is to a tube) to determine the outer wall and then used centerlines to selectively remove the regions that are of no interest and finally levelset based method for final geometry reconstruction and tracing the dissection wall. Similarly Kovács et al. (2006) have utilized Hough transform, a technique for isolating features of known shapes (in this case, circles) within an image, to delineate the outer

wall of the dissected aorta and used a measure of sheetness (a measure that indicates how similar the local structure is to a sheet), utilizing Hessian matrix for detection of dissection wall. Recently, Dillion-Murphy (Dillon-Murphy et al. 2015) applied Arterial Spin labelling (ASL) techniques of segmentation in 2D paradigm for tracing contours of true lumen and false lumen. ASL is a technique in which magnetization of arterial blood water is labeled by magnetic inversion or saturation, and the delivery of labeled blood water is observed (Wong 2014). These contours were then be lofted together using non-uniform B-splines (NURBS) for generating smooth 3D model required for flow simulations. Other researchers (Rinaudo et al. 2014; Shang et al. 2015;Chen et al. 2013) have used commercial software VMTK, Amira, and ShIRT for 3D reconstruction of aortic dissections for studying hemodynamics. No further detail regarding the pipelines followed were given by these groups.

Since our starting point is CT scan of a patient with dissected aorta, and we want to obtain a levelset representation of the diseased aorta, in current work, we closely followed the algorithm developed by Krissian et al. (Krissian et al. 2014), but not completely. The end goal of the algorithm followed by Krissian et al. was extracting the dissection wall, and their method included tedious manual steps for the removal of centerlines of the branches that were not necessary. Their algorithm also included vague steps like performing hysteresis thresholding until the desired intensities were enhanced. Hence we improvised their algorithm to make it more semiautomatic and suit our needs.

We took a three step approach for segmentation of dissected aorta. In the first step, the outer wall of the aorta (comprising of true lumen, false lumen and dissection wall) is segmented as a single object of interest. In the second step the dissection wall is segmented and in the third step the dissection wall is removed from the outer wall segmented during step one and the resulting entity is then further segmented to obtain a smooth levelset representation of the dissected aorta. In this work, existing techniques have been combined in a pipeline to enable a semi-automatic segmentation of the dissected aorta. The general scheme of the approach is illustrated in Figure

23. This method will be explained in further detail below, and is referred to as Method III in this work.

4.5.1 Extraction of outer wall

As mentioned the section above, we define the outer wall for a dissected aorta as a combination of all the voxels representing the true lumen, false lumen, dissection and tears. In order to extract the outer wall, region growing approach of segmentation of aorta, explained in section 4.4, is followed. The resultant image obtained from the pipeline is a levelset field (see Figure 27 (a), (b) & (c)), which still contains the dissection wall. To remove this wall, a binary thresholding is performed to create a binary image where all the voxels less than and equal to the zero levelset (i.e. those voxels inside of the outer aortic wall region) are assigned an intensity of one, while all other voxels are assigned intensity of zero. Zero levelset representation of outer wall is shown in green in Figure 27 (c) and an image of the resultant binary image superimposed over the original intensities is shown in Figure 27 (b) for two dissection cases. Binary image thus obtained will be used below to extract the dissection wall.

4.5.2 Extraction of dissection wall

General flowchart for extraction of dissection wall and Method III is given in Figure 23. A more elaborate flowchart showing the images resulting from individual operations are shown in Figure 34. The binary image BI (Figure 34 (2)) obtained from the segmentation of the outer wall is used as a mask to extract the original intensities of the true lumen, false lumen, dissection and tears from the original scan, this resultant masked image (Figure 34 (3)) will be referred as M_I . In general, we can see that the highest intensities in M_I belong to true lumen, lower intensities belong to false lumen and the dissection wall is characterized by the lowest intensity. However, the intensity range of the true lumen, false lumen and dissection wall are not consistent throughout M_I . They vary in different cross section depending on the location of the tears. A close observation of the individual cross sections show that the local intensity distribution follows the global intensity (see Figure 27 D1 (a)) pattern of highest intensities in true lumen and lowest in the

dissection. Hence we performed a set of statistical operations in a slice at a time to delineate the dissection wall. These steps are explained in the paragraph below. After the final step of extraction of pixels belonging to dissection wall, same statistical operations would be carried out in the next slice. A pseudo-code for the steps below is given in right inset in Figure 23.

First a local weighted average ' m ' was calculated for all non-zero intensities in a contour in individual slice as shown in Figure 34 (3). Now a labeled image ' L ' is created where all the non-zero pixels in M_I that are less than m are labeled as ' FL ' and rest of the non-zero pixels are labeled ' TL ' as shown in Figure 34 (4). In order to eliminate any outliers in contours median filtering is done in for each individual slices in L and we obtain a new image ' L_m '. Now weighted average ' m_{FL} ' is calculated for all the pixels in the slice with label FL in image L_m . Gradient magnitude, calculation of change in intensity of the image intensities along axes, is then computed for L_m resulting in an image ' L_{GM} ' as shown in Figure 34 (5). In this figure we can clearly see the pixels defining the boundary of the outer wall have higher intensity than the pixels that define the dissection wall and hence appear brighter. Now all the voxels with less than 50 gradient magnitude and intensity less than m_{FL} are labeled as dissection wall ' DW ', these pixels are shown in Figure 34 (6). Images of full dissection wall or septum so extracted from two dissected aortas are shown in Figure 27 (d). Extraction of the dissection wall is then followed by the segmentation of dissected aorta which is elaborated below.

4.5.3 Segmentation of dissected aorta

For all the voxels labeled as DW in L_{GM} (see Figure 34 (5) and (6)), the corresponding voxels in BI (Figure 34 (2)) are assigned zero intensity. Hence, the resulting binary image ' $B2$ ' as shown in Figure 34 (7), of a dissected aorta is obtained. A levelset representation of the dissected aorta is then created by using two step levelset based segmentation approach as explained in section 4.3.7. In Figure 27 (e), we can see the zero levelset representation of the final segmentation. Since the outer wall has already been segmented using geodesic active contours and hence made smooth, a propagation, advection and curvature scaling values of 1 was found enough for this step.

However, users should experiment with different scaling values to get the best result. Users should note that excessive smoothing or ballooning at this point may result in collapse of dissection wall hence proper care should be taken while increasing the values of these scaling parameters.

4.6 Validation of Method I (Objective 1) using phantom model of ICA

Different methods of segmentation have different degrees of accuracy which depend hugely on the quality and source of the image being segmented, as well as on the algorithm itself. Accuracy estimation and segmentation algorithm validation is especially challenging when segmenting an *in vivo* image because there is no alternative means to physically measure the object of interest. Expert human “raters” and automated algorithms are consequently considered the accepted measure of evaluating the success of a segmentation procedure, though both of these routes have limitations (Warfield, Zou, and Wells 2004). Ideally, algorithms are first validated using digital or physical phantoms, which are created from known geometries and pre-defined measurements that can be compared with that of the segmented image. Since our goal is to validate the segmentation procedure, we first carried out tests using an image source with known physical measurements. In this study, a patient-specific image of an intracranial aneurysm was segmented using geodesic active contour based segmentation algorithm (i.e. Method I). To validate this geodesic based segmentation method, we segmented the CTA image of an anthropomorphic Intracranial Vascular flow phantom that contains Circle of Willis and a large saccular aneurysm originally obtained from Shelly Medical Imaging Technologies Ontario, Canada. This phantom, shown in Figure 12, was imaged at University of Iowa Hospitals and Clinics (UIHC) using Siemens Sensation 64 CT scanner, (Siemens Medical Solutions, USA), following the protocol for scanning real aneurysms (S. I. Dillard et al. 2014). An image of dimension 512x512x249 was obtained with pixel spacing of 0.3086 mm in x and y direction and a gap of 1 mm between the slices in z direction. The most challenging aspect of the dicom image of the phantom is the low contrast between vessel and background. It is possible to check the accuracy of the segmentation algorithm by comparing it with the manufacturer’s specification of the phantom

A comparison between measurements of phantom and final result of current segmentation method are given in Table 1 below and a qualitative comparison can be seen in Figure 16. The diameters of the bulb of the aneurysm in x and y planes measured 10.74 and 10.61 mm respectively with 7.4% and 6.1% error. The length of the bulb in the z plane from the top to the point where the neck starts is measured as 13.058 mm with 0.44% error and the diameter of neck is measured as 3.5 mm with 12.5% error. The diameter of the left anterior cerebral artery from current method is measured to be 2.8 mm with 6.6% error. If we consider the spatial resolution of image, the maximum error is observed in bulb diameter which is of the order of about 2 pixels. Overall the results of the current segmentation method compares closely with that of previously employed levelset based segmentation method (S. I. Dillard et al. 2014) which is referred to as Method 0 in this work.

Table 1 Comparison of Sigmoid-GAC based segmentation method (Method 1) with Method 0.

Diameters compared	Phantom	Method 0		Method I	
	Diameter (mm)	Diameter (mm)	% Error	Diameter (mm)	% Error
Bulb (Xaxis)	10	9.33	6.1	10.74	7.4
Bulb (Yaxis)	10	9.44	5.6	10.61	6.1
Bulb (Zaxis)	13	13.1	0.77	13.058	0.44
LAC	3	3.04	2.4	2.8	6.6
LeftBigArtery	5.5	X	X	5.75	4.5
Neck	4	4.06	1.5	3.5	12.5

4.7 Validation of Method II (region growing based segmentation)

Phantom image of ICA is also used as the ground truth to validate Method II, a region growing based segmentation method. The comparisons made in table below show maximum error of 3.6% in left anterior cerebral artery, which is still less than 1 pixel. If we consider the spatial resolution of 0.3 x 0.3 x 1 mm, then error in all the cross sections compared is of the order of less than one pixel. Comparatively Method II is more accurate than Method I and 0.

Table 2 A Comparison of region based segmentation method with Method 0.

Diameters compared	Phantom	Method 0		Method II	
	Diameter (mm)	Diameter (mm)	% Error	Diameter (mm)	% Error
Bulb (Xaxis)	10	9.33	6.1	10.3	3
Bulb (Yaxis)	10	9.44	5.6	10.3	3
Bulb (Zaxis)	13	13.1	0.77	12.3	1.9
LAC	3	3.04	2.4	2.88	4
LeftBigArtery	5.5	X	X	5.7	3.6
Neck	4	4.06	1.5	4	0

4.8 Validation of Method II and Method III using overlap fraction (OF)

Region based segmentation method was formulated for extracting anatomical features like aorta which have other unwanted entities of comparable intensities at close proximity. Since we did not have access to any phantom model of physiologic regions including both the aorta and surrounding tissue, we compared the results of the current region based segmentation method (Method III) with the result of manual segmentation done in Mimics v13.0 (Materialise, Leuven, Belgium). We calculated the overlap fraction (OF) between the two segmented entities in the

image domain to check the spatial agreement between Method III developed here, with the commercially available algorithms that have been used for aortic dissection segmentation by previous researchers (Tse et al. 2011).

Where,

$$OF = \frac{manual \cap region\ based}{\min(manual, region\ based)} \times 100 \% \quad (8)$$

Here, *manual* and *region-based* are counts of voxels from manual and region based segmentation methods. CT scans of two healthy aortas (“P1” and “P2”) and two dissected aortas (“D1” and “D2”) were used for this validation. P1 is an image of dimension 512x512x108, with a spatial resolution of 0.769531 mm in x and y directions and a gap of 3.0 mm between the slices in z direction. Similarly, P2 is an image of dimension 323x323x100 with pixel spacing of 0.630569x0.630569x3.0 mm. D1 is an image of a dissected aorta of size 512x512x211 and spatial resolution of 0.82226x0.82226x3.0 mm and D2 is an image of a dissected aorta of size 218x209x84 and spatial resolution of 0.683594x0.683594x3.0 mm. We can see the calculated results of *OF* for healthy and diseased aorta in Table 3. Good overlap of 97% and 99 % were obtained for healthy aortas P1 and P2. Overlap of D1 is 100%. Similarly *OF* for D2 is also 100%.

Hence, it can be concluded that results of Method II are in good agreement with both ground truth as well as gold standard. Similarly Method III is also in good agreement with the ground truth. However, the results are underestimated when compared to gold standard.

Table 3 Overlap fraction (OF) between manually segmented and region based segmentation results for patient specific images of aortas P1, P2, D1 and D2. High overlap fraction is indicative of high spatial agreement between volumes.

	Voxel count			
	P1	P2	D1	D2
Manual	262858	117445	91920	201800
Method II or III	271297	96152	85071	166794
only Manual	7511	21548	6849	35016
only Method II or III	15950	225	0	0
Intersection	255347 (97%)	95897 (81.65%)	85071 (92%)	131768 (65.3%)
Union	278808 (1.06%)	117670 (100.2%)	91920 (92.5%)	166794 (82.7%)
OF	97%	99%	100%	100%

4.9 Conclusions on image segmentation capabilities

In this chapter, we talked about the three image segmentation algorithms Method I, Method II and Method III. All the three algorithms have different image preprocessing pipelines but a common final segmentation step, geodesic active contour based levelset segmentation, for the final levelset generation. While both Method I and Method II gave good results for segmentation of the phantom, Method II was closer to the ground truth when applying same levelset scaling parameters (maximum error was about 2.5 pixels for Method I and less than 1 pixel for Method II), and hence more accurate. It should also be noted that the usage of these two methods are interchangeable in general, but Method II is more robust when the region of interest is in close proximity of features with similar intensities. Finally, a new pipeline, Method III, was developed utilizing the pre-existing techniques for the segmentation aortic dissection. Good overlap fraction was obtained when comparing the results from Method III with the gold standard.

CHAPTER 5. FROM LEVELSETS TO FLOW COMPUTATIONS

5.1 Objective 2: Ensure image to flow computation

No matter how robust or accurate a method or a tool is, it is not useful or successful until one can use it to solve problems holding significance. The ultimate goal in developing this toolkit is to enable the analysis of fluid dynamics in biomedical systems by filling the gap between medical images and flow calculation in a user friendly setup without mesh generation. The first few steps towards this goal was taken by performing segmentation which was followed by hemodynamic analysis of the following biomedical case studies. The main goal of this exercise is to bridge the gap between biomedical images and the corresponding flow systems. Hence, the second objective is to ensure the successful execution of the whole mechanism of going from images to solving flow without the need of mesh generation with the help of our toolkit and in-house flow solver.

Pain-staking grid generation involved in patient-specific modeling indicates that it's a wise step to boot out the task of generating mesh and find a simpler alternative. Our objective was to show that such simple route exists and is equally accurate and more feasible. Our approach involves combining levelset based methods widely used for segmentation purposes with levelset descriptions of embedded boundaries in Cartesian grids. The binding element between the two being the levelsets which can be obtained as outcome of several segmentation methods and it also supports sharp interface based flow calculations.

After recognizing the challenges involved in the conventional route of image-based CFD, that involved body fitted mesh, a fraction of researchers' pursued an alternate route of solving the equations of fluid flow in complex boundaries using non-body fitted mesh, where they dealt with the challenges regarding imposing boundary condition by either modifying the scheme used to discretize flow variables near boundaries (Udaykumar et al. 2001) or by forcing terms (Peskin 1972; Bagchi, Johnson, and Popel 2005) that are active in the vicinity of the immersed boundaries. Immersed Boundary Method (IBM) (Peskin 1972), which was used to model the effect of an elastic

material on fluid motion was the first of these approaches. In a way IBM mimics embedded boundaries, as it acquires volumetric body forces that are needed as source terms in Navier Stokes equations by using numerical delta function to transform physically based surface force (which are available). Though IBM has enjoyed much success in modeling flexible structures at low Reynolds number, it falls short when it comes to modeling rigid objects (Leveque and Li 1994) and moderately high Reynolds number flow (Lai and Peskin 2000; S. I. Dillard et al. 2014).

In order to make Cartesian grid based methods most robust in addressing boundary problems, two varied approaches were considered. The first being discrete forcing approach and the second being sharp interface approach. The discrete forcing approach is one in which the physically based forcing term used in original IBM methods are replaced by forces derived from numerical schemes themselves, for instance, velocity boundary conditions that are satisfied at prescribed locations. The second approach where the interface is treated sharply involves incorporating boundary conditions directly into discretization operators near fluid-solid or fluid-fluid interface. Examples of this type of approach are the Immersed Interface Method (IIM) (Li and Lai, 2001), the Ghost Fluid Method (GFM) (R. Mittal et al. 2008; Tseng and Ferziger 2003), the Sharp Interface Method (SIM) (Udaykumar et al. 2001), and the Extended Finite-Element Method (XFEM) (Chessa and Belytschko 2003).

In order to demonstrate the feasibility of image based modeling of complex geometries by using Cartesian grid based methods, it is necessary to obtain the information about embedded boundaries in a form that is useful to these methods. This is where levelset methods come into play. Not only do levelset field intrinsically define embedded boundaries as zero isocontour of a higher dimensional signed distance function (Sethian 1999), but they are also the basis of several segmentation algorithms. Since images are voxelized representation of intensities where the voxels with gradient defines object boundaries, one can easily view them as intensities arranged in some kind of Cartesian grid and hence find the analogy between the two. Thus, many segmentation algorithms have been built based on levelset methods and they have proved their metal in generating smooth segmentation surfaces (Suri et al. 2002). Since smoothness is one of

the key factors in accurate representation of physical geometries and in solving the equations of fluid flows, levelset based segmentation method have seen wide acceptance in the field of medical imaging physics (Angelini, Jin, and Laine 2004; Suri et al. 2002).

In order to validate the results obtained, segmentation was carried out in a patient-specific CTA image of ICA and flow was solved in pELAFINT solver. These findings have been compared with the previous study done by Seth Dillard on the same image using active contour algorithm. More on this can be found in section 5.8.

5.2 The levelset method

Levelset field is a signed distance map where the object boundary is denoted by zero contour and points inside the boundary and represented as negative distance while those outside of the boundary are denoted by positive distance as shown in Figure 5. As the levelset based segmentation method is incorporated, this zero contour also becomes the representation of embedded boundary. Levelset representation is hence the binding element of our image based modeling framework. It inherently bridges the gap between image segmentation and Cartesian grid based flow solver.

There are several advantages of levelset methods. For one, it is easy to determine a region belonging to an object based on the sign, secondly it is convenient to extract geometrical information about the embedded object like normal vector and curvature by performing simple finite difference operations on the Cartesian mesh, i.e. if Φ be the flux, then normal vector and mean curvatures are respectively given by $\vec{n} = \nabla\Phi$ and $\kappa = \nabla^2\Phi$. These information hold high significance in flow calculations as normal vectors are important for constructing operators required for supplying Neumann (flux) boundary condition and to determine the location where boundary conditions need to be applied in Cartesian grid solver (Mousel 2012). The other advantage of using levelsets is that arbitrarily complex shapes can be modeled and topological changes such as merging and splitting are handled implicitly. Levelset models are topologically flexible and can split and merge as necessary in the course of deformation without the need for re-

parameterization. Using Osher and Sethian levelset approach, complex curves can be detected and tracked and topological changes for the evolving curves are naturally managed.

5.3 Deriving levelset information to supply to the flow solver

Quality of segmentation directly affects the outcome of a CFD simulation, so for accurate flow results it is essential that the object of interest has been captured as accurately and smoothly as possible. Intensity based methods have been found to give smoother segmentation than other cluster based methods (S. Dillard et al., 2014.) levelset based method falls into the intensity based method category. Geodesic active contour based segmentation method has been chosen to extract levelset representation of our ROI as described in section 3.3 such that the zero contour of the levelset represents the embedded boundary/surface of the object of interest.

5.4 Flow solver with embedded boundaries

Substantial amount of effort has been put forward by our Thermo-fluid group in developing a robust flow solver for solving transport phenomena in complex domains (with stationary and moving boundaries) based on fixed Cartesian grid Eulerian framework (Vigmostad et al. 2010; Udaykumar, Krishnan, and Marella 2009) and addressing the challenges involved while developing techniques for treating the embedded boundaries as sharp entities (Udaykumar, Krishnan, and Marella 2009; Marella et al. 2005; Sambasivan and Udaykumar 2011).

5.5 Cartesian grid method

The ultimate goal of building user interactive tool kit is to compute fluid motion in geometries defined by levelsets extracted from images. The flow field is obtained by solving the incompressible Navier-Stokes equations:

$$\nabla \cdot u = 0 \quad (9)$$

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = -\nabla p + \frac{1}{Re} \nabla^2 u \quad (10)$$

Where, u and p are the fluid velocity vector and pressure respectively, and $Re = UL/\nu$ is the Reynolds number. U and L are characteristic flow velocity and length scales, and ν is the kinematic fluid viscosity. The governing equations are advanced in time using a 2nd-order four-step fractional step algorithm to segregate the pressure from the velocity (Mousel 2012). The non-linear and viscous terms are integrated temporally using the 2nd-order explicit Adams-Bashforth and semi-implicit Crank-Nicolson schemes, and all spatial derivatives are approximated using 2nd-order central differencing. The variables are stored at mesh cell centers with the exception of a set of cell face velocities (satisfying the divergence-free condition over each mesh cell) that are used to construct the advection terms.

Boundary conditions on embedded boundaries (defined by the zero-contours of the embedded narrow-band levelset field) are applied using a least-squares formulation of the sharp interface method. In this approach, a three-dimensional quadratic least-squares method is used to extrapolate flow variable values to a set of ghost nodes immediately outside the flow domain (Mousel 2012). The least-squares formulation allows one to obtain locally 2nd-order spatial convergence rates on arbitrarily shaped domains. The implicit filtering inherent to least-squares methods also enhances the robustness of the approach when geometries with noisy surfaces are simulated. A detailed analysis of the current interface treatment is described in (Mousel 2012).

5.6 Mesh structure

Analyzing internal flow through biomedical geometries means working with highly tortuous geometries with lots of branching. Using normal Cartesian grid to represent this geometry would mean using lots of unnecessary grid points to represent a small geometry by virtue of its cuboidal shape. Further, flow calculations only take place in the internal domain of the geometry where actual flow occurs so this means waste of huge memory at each time step. Since parallel algorithm has been opted for flow solver, such situation could also lead to huge load imbalance. Because of these reasons and to save computation time current flow solver operates on a Cartesian *base mesh*, initial mesh without any refinement, equipped with the capability of local mesh

refinement and *pruning* (trimming) of mesh (Mousel 2012). The extraneous mesh that fall outside of flow domain and *narrow band*, tube of rid points that move with interface, are subjected to get discarded during the set up phase of the simulation. Following this screening, local mesh refinement is carried out on the remaining cells based on the forest-of-*octrees*, a tree data structure in which each internal node subdivides into eight children, approach (Burstedde, Wilcox, and Ghattas 2011), where each base mesh cell holds the root of a fully-threaded octree, and the nodes of the octree with no children correspond to fully refined mesh cells.

For instance, in the mesh refinement resulting from a simulation in a 90 degree bend case observed in observed Figure 13, it can be clearly seen that the mesh considered for solving flow is of the shape of the bend tube itself with mesh refinement imposed in a small band around the tube surface. In order to facilitate the construction of discrete operators, the mesh satisfies a 2:1 balancing constraint where no mesh cell has a neighboring cell across a face, edge, or corner that is more than one level coarser or finer. The 2:1 balance constraint is imposed through the use of the highly scalable Prioritized Ripple Propagation algorithm (Burstedde, Wilcox, and Ghattas 2011).

5.7 Solver validation

Over the years pELAFINT flow solver has been extensively validated against different cases for varied range of Reynolds number (Mousel 2012) and has demonstrated its capability of capturing all the relevant physics. The cases that have been used for validation include deformation of a circular levelset by a vortex flow, the Zalesak disk problem (Zalesak 1979), impulsive flow past a stationary cylinder against, flow past a stationary sphere (Marella et al. 2005; Rajat Mittal and Najjar 1999; Johnson and Patel 1999), flow in a bent tube (Van De Vosse et al. 1989), flow in a vocal fold geometry (Chisari, Artana, and Sciamarella 2011), flow past an impulsely started moving cylinder (Koumoutsakos and Leonard 1995), sedimentation of sphere (Johnson and Patel 1999) and others.

The most relevant validation according to the type of flow computation required and for this work is the one with the work of Van de Vosse in 1989, where he conducted experiments using Laser Doppler Velocimetry to calculate velocities at center plane of the 90 degree bent tube. This case was considered for validation because of the challenges involved in capturing fine vortices and secondary flows accurately. Same case containing a 90 degree bent tube with a unit diameter and a curvature of 6 fold the diameter was emulated. Outlet was extended by a length of a diameter. A fully developed parabolic flow was supplied at the inlet and Convective boundary condition was supplied at the outlet; Reynolds number of 300 was maintained so that same flow conditions could be simulated. Cartesian mesh considered contained pruning capability but was devoid of local mesh refinement. Instead a small grid spacing of 0.02 was maintained because of prevalence of vortices in the flow. When the centerline velocities were compared, they were found to be in total agreement with the experimental results of Vossee which has also been presented in Table 1 and Figure 13. The agreement lead to validation of newly added mesh pruning capacity in addition to the computational advantage it has added.

5.8 Validation of image-levelset-flow setup (flow in cylinder and intracranial aneurysm)

After the validation of the segmentation capability, next progression was to apply the method on a real data set and check if the flow simulations for the segmented geometry from the images give acceptable results. For this purpose, we conducted two studies: first we chose a simple cylinder and evaluated the length required for the development of fully developed flow. We compared the result with empirical solution given by following equation.

$$l_e = 0.05 \times d \times Re \quad (11)$$

Where, l_e is entrance length, Re is Reynolds number and d is the diameter of the cylinder. For this purpose we created an image of a cylinder of 2 m diameter and 20 m length as shown in Figure 40. The dimensions of the image was 5x5x20 with the spatial resolution of 0.05x0.05x0.05. The image was segmented using geodesic active contours segmentation method to obtain levelset representation of the cylinder. Flow was simulated for a Reynolds number of 50 in our in-house

flow solver. Velocity vectors at different lengths along the cylinder were checked and the result was in agreement with the empirical solution of 5 m. 3d and 2d views of velocity vectors plotted at different lengths are shown in the Figure 40. In the bottom most figure, length scale is also shown for convenience of checking the entrance length.

In the second study, we aimed to obtain an accurate levelset representation of a 3D CTA image of ICA (See Figure 1) using our toolkit. The CTA data set was acquired at UIHC with a Siemens Sensation 64 scanner which consisted of 319 512×512-pixel slices, spaced 0.5 mm apart, with pixel dimensions $\Delta x = \Delta y = 0.390625$ mm. As mentioned earlier, the main target of developing this toolkit is to obtain smooth transition from images to fluid dynamics. So, this study was furthered by conducting fluid dynamic analysis in our in-house solver on the levelset field so obtained. The accuracy of the flow field obtained in this manner was then compared with the results obtained by simulation of the same geometry with same boundary condition in FLUENT v14.5.0 (ANSYS Inc., Canonsburg, PA). Steady state simulation was carried out with blood as fluid of density 1060 kg/m^3 and viscosity of 0.0035. Inlet velocity of 0.3657 m/s and 0.336 m/s were supplied at inlets 1 and 2 and mass split of 50:50 was assigned at the two outlets as shown in Figure 1

We can see good agreement when comparing the results of pressure and velocity vectors in Figure 41 and Figure 42 respectively. Pressure at inlet1, inlet2, outlet1 and outlet2 are in the range of 1800 Pa, -2000 Pa, 1000 Pa, 200 Pa for both our solver and FLUENT. Similarly the overall range of velocity vectors looks similar with the velocity magnitude of 2.0 and 0.5 m/s in outlet1 and outlet 2 and lower velocities of about 1 m/s in the outer region of bulb due to decrease in pressure. However the magnitude of velocity at the junction of inlet1 and bulb is 3.7 in our solver while it is 3.2 in FLUENT. This difference is because cells with higher velocities are refined in our solver.

5.9 Conclusions on from levelset to flow simulation

In this chapter we talked about our framework for going from levelset representation of geometries to flow simulations. Validation of flow simulations for flow in a cylinder case showed good agreement for the entrance length when compared with the empirical result. Similarly, when the result of steady state flow simulation in ICA was compared against the simulation in FLUENT, good match was obtained. We noted that results from our flow solver were more refined compared to that of FLUENT.

CHAPTER 6. MANIPULATION OF SEGMENTED OBJECTS

6.1 Objective 3: Add general object manipulation capabilities

In some cases, it may be required to modify or manipulate the object in the image that has been segmented. For instance, in order to analyze the effects of branch flow or curvature in the blood vessels, it may be necessary to add or remove branches from an artery. Similarly, generally researchers have images of diseased cases like of a vessel with aneurysm, but not of the vessel condition prior to aneurysm. Then if one wishes to analyze the effect of aneurysm one would certainly want to remove that aneurysm from the vessel or vary the size of the aneurysm. Further, in most vascular researches it is desired to extend vessels at their extremes by a length of at least 5 diameters in the direction normal to the flow so that the flow results would be devoid of any end effects (Shrestha 2010; X. Y. Xu et al. 2012). Furthermore, to apply certain boundary conditions like Dirichlet or Womersley profile, the inlet and outlet the shape of the boundary needs to be circular which is highly unlikely in patient-specific scenario. For accomplishing circular inlet, the extensions of the vessels need to be morphed which also requires manipulating the segmented entity.

In order to fully understand a phenomenon, it is often required to isolate the individual effects of different geometric parameters for which it is necessary to manipulate the geometry. Similarly, in order to apply appropriate boundary conditions like Dirichlet or Womersly at the inlets or in order to alleviate the end effects in the flow simulation it becomes necessary to manipulate geometry. Different softwares like Auto-cad, Pro/E, Gambit, ANSYS workbench etc. are available in market which offer capabilities of object manipulation for regular geometries. However, when it comes to real objects with irregular boundaries particularly patient-specific models, such softwares fail to create any impression. This is also one of the reasons why researchers still do not delve in to patient-specific modeling though they fully understand its significance. Though few newer toolkits like VMTK have put together few options for object manipulation they are still case specific like for vascular geometry and aneurysm for VMTK.

Hence, our third objective is to develop a toolkit which is not only capable of segmenting images but is also capable of manipulating the segmented objects. For instance: ability of extending the inlets and outlets, adding or removing branches, smoothing the region of aneurysm from vessel etc.

6.2 Conventional route of object manipulation

Conventionally, manipulation of the segmented object is done on the surface representation of the region of interest. Marching cubes is by far the most popular algorithm for iso-surface extraction of the segmented geometry (Dietrich et al. 2009; Lorensen and Cline 1987). If such surface representation is deemed sufficient of flow analysis, a volumetric representation of the surface is created by chopping the tubular ends of the closed surface and capping them and finally creating a volumetric mesh representation of the closed region. In many cases the extracted surface may require some general manipulation.

Extension of the ends of vessel is a commonly sought after manipulation for objects segmented for CFD purposes. This operation is performed after chopping off the bulb like ends of segmented entity. Different approaches have been taken in past when creating extensions at the ends of the vessels. One approach used in meshing software like Pointwise, Workbench etc. is supplying the direction and length for extrusion of the inlet and outlet faces. Another approach found in CAD and meshing software is setting the direction of extrusion to be normal to the boundary faces. Similar approach is found in VMTK, where the centerline of the vessel and the direction normal to the cross section to be extruded is calculated. Another simple approach is to continuously copy the boundary pixels until the desired length of extension is obtained by dilation.

Clipping of a surface is hence a necessity for conventional model generation for CFD and also if extension of inlets and outlets is desired. Though CAD softwares have this capability for ideal geometries, it is almost nonexistent for patient specific geometries. VMTK, IAFEMesh (for orthopedic purposes) and other toolkits based on VTK offer this capability. Though clipping capabilities are offered in both the toolkits, the plane selection is arbitrary and totally based on the

judgement of the user. A user will try to cut the vessel such that the cutting plane looks normal to the vessel at that point. However, a cut in a normal direction is almost never achieved. We have addressed this situation by developing a capability that cuts the geometry in exact normal direction utilizing the calculated centers of the object.

Removal of entities is a capability that may be desired for purposes like removal of small vessels from segmentation. One may choose to do this either before or after segmentation. In image domain one can put a marker in the vessel that is not to be segmented out and hence avoid the vessel from being segmented (Luca Antiga and Steinman 2006; Nanduri, Pino-Romainville, and Celik 2009). If it is done in surface domain, the unwanted vessels can be trimmed off by using a clipping plane. However, the later method though possible will result in kinky and irregular surface/mesh where the cut is made. Such a surface is problematic for CFD purposes mainly because it results in a bad quality volumetric mesh and hence erroneous solution. Removal of entities may be useful in cases where effects of certain geometrical feature is being studied. For instance effect of aneurysm on hemodynamics of blood vessel, effect of head vessels on aortic flow.

6.3 Marching cubes method

The marching cubes algorithm (Lorenson and Cline 1987) was originally developed by Lorenson and Cline and published in 1987 Siffraph Proceedings. Two dimensional algorithm for extracting an isoline for a specified value is known as marching squares where for each vertices of a square one has 2 choices, resulting in 16 (2^4) possible cases of an isoline passing through a square. The extension of this method in 3d is known as marching cubes. In three dimensions there are 8 nodes to be considered and hence 256 (2^8) possible cases of how an isosurface can be passing through a cube. There are cases of complimentary symmetry and rotational symmetry and few are even ambiguous cases out of those 256 cases.

There are again 256 cases for a voxel since each value at the 8 voxel vertices can either be above or below the isosurface value. The common approach to implementing the marching cubes

algorithm is to generate a 256 entry lookup table (LUT) in which each entry specifies the triangles for that specific case. Out of 256 cases, 128 of them are complimentary symmetry, 14 are rotational symmetry, and in actual they can be summed up by 15 topologically unique cube combinations. So another approach is to store 15 topologically unique cases and information about the triangles for each of the 15 cases as shown in Figure 17 and use this information to generate the 256 entry lookup table.

Each triangulation contains 0 to 4 triangles and can be stored as a list of triangles where each triangle is a list of 3 numbers which are actually indices of the edges on which the vertices of the triangle lie (See Figure 17). The vertex location for each triangle is calculated by interpolating the data values at the two vertices where one is above and another is below the specified isovalue. The normal for each triangle can be determined by calculating the cross product of the two edges and then normalizing the result. On the other hand, vertex normal can be calculated by summing up normal of all the triangles that are using that vertex and then normalizing the total.

Marching cubes algorithm:

Compare the value at each vertex of each cube with the designated isovalue. This will determine if the vertex is inside or outside of the surface. For each vertex which lies inside of cube, a bit is inserted into the index (cubeIndex).

The edgeTable array is indexed using cubeIndex from step 4. That in turn returns a number, bits of which represent the 12 edges that intersect the cube (there are 256 possibilities). Zero is returned if all are inside or outside.

If the cube is intersected by the surface, a point of intersection will be assigned for edge that is intersected as determined by the bits in the number returned from edgeTable (using Linear Interpolation). All of the points are put in an array of size 12, each representing its edge.

CubeIndex is used again with triTable which returns indexes for the array of points defined in step 6. triTable is a double 256x16 array, representing 256 possibilities of intersection and up to 15 vertices for the triangles formed in each possibility. The points are in order, so for example triTable will be the index of the first vertex of a first triangle of the third possibility, and so on.

Those indexes are used with array defined in step 6 to group vertices in appropriate triangles. Normal vector is defined as the cross product of two vectors for each triangle.

6.4 Object manipulation capabilities added in IAFEMesh

In this work, we have built object manipulation capabilities that are essential for modeling biological flows for CFD. These capabilities, enlisted in Table, are useful for patient specific model generation irrespective of the flow solver used. The main difference between manipulation capabilities developed in this work and in other ITK-VTK based toolkits or CAD/meshing software is that in this work all the manipulations to the object of interest is done in image domain. As mentioned earlier, surface manipulation capabilities offered for patient specific geometries in commercial software are very limited. And the capabilities that are offered, were developed for idealized geometries. A user in need of manipulating a patient specific geometry, thus ends up jumping from one software to another and using multiple software to obtain a decent manipulation. For example, if a user is successful in performing a Boolean addition of two surfaces, it will be too difficult of a task to smooth out the region of intersection. Sharp transitions are neither characteristic of biological flow nor are they a good for a CFD model. Such difficulties in obtaining a realistic Boolean operation or smoothing out the sharp transitions led to the inception of attempting manipulations in image domain.

With the purpose of testing and demonstrating our object manipulation capability within IA-FEMesh, a study was carried out on the role of head vessels in the ascending. Only the capabilities incorporated in IA-FEMesh for object manipulation purposes will be discussed in this section.

6.4.1 Center calculation

Determining centerline of the segmented object is very crucial in this work as it forms the basis for operations like clipping the object in normal or chosen direction, extrusion of the object in normal or chosen direction, adding of entities and for post processing purposes. Centerlines can be obtained by thinning an image or by creating a signed distance map of the image. In this work,

binary thinning is used for find the centerlines (“skeleton”) of objects in the input image. The general idea is behind binary thinning is to erode the surface of the object iteratively until only the skeleton remains. Erosion should be performed symmetrically in order to guarantee the medial position of the skeleton lines and such that the connectedness of the object is preserved. There should not be any holes or cavities in the resultant object. There are two major approaches to thinning an image: kernel based methods and decision trees. Kernel-based filters apply a structuring element to the image and can generally be extended to dimensions higher than 3D, to find computationally efficient solutions for 4D and higher dimensions is subject of ongoing research. Methods based on decision trees are thus far limited to 2D and 3D, but are potentially faster than morphological filters, if they are well designed and can find more deletable points at each iteration.

Kernel based methods cannot handle all the 226 possible binary combinations of object and background voxels in a 26-neighbourhood framework. Lee et al. have demonstrated that a decision tree based method can handle these cases as well as correctly determine all the possible deletable points (Lee, Kashyap, and Chu 1994).

Following tests are performed in order to check if a pixel can be eroded from the object. This is carried out for all pixels in the image and repeated until there are no more changes.

1. If the current pixel is a surface pixel. This test only considers one the six possible directions in 3D at a time in order to perform thinning symmetrically, i.e. to guarantee that the centerlines are not shifted to one side of the object.
2. If the pixel is not the end of a line, because the center lines are to remain.
3. If the deletion of the point would not change the Euler characteristic, i.e. if no holes are created when deleting the pixel. This is done by using the look-up table (LUT) from (Lee, Kashyap, and Chu 1994).
4. If the current point is a simple point that is its deletion does not change the number of connected objects. This test is performed last as it’s the computationally most expensive one.

The output of the filter is a binary image where center points form the foreground pixels. These centers points are available for picking or selecting by the users when selection mode is active. In such modes, these points are coded to appear as green spheres. Users are allowed to right click on the sphere to be selected and on selection the color of the sphere changes from green to pink.

6.4.2 Extraction of oblique plane

In order to analyze/compare cross section as a result of segmentation, check the flow features while post processing of CFD results, or extrude cross section of a segmented object along a normal or any specified direction, a capability of extracting oblique section is necessary. An oblique section of an image is an arbitrary 2-D plane through a 3-D Image. Such oblique section can be obtained if the center point of the plane, a direction to look at is known and a second vector is known. The class `itkLookAtTransformInitializer` developed by Mueller (Mueller 2007) directly computes the underlying 3×3 matrix, center point, and translation. The matrix is specified by finding the axis orthogonal to the direction and up vectors. The center of the transform is the center point specified by the user. The translation is computed using the center of the image. The `itkObliqueSectionImageFilter` encapsulates the transform initializer, resampling, and extraction. The internal `itkResampleImageFilter` is configured to produce a one slice 3-D image and the internal `itkExtractImageFilter` is used to extract the final 2-D image. User has the ability of selecting the interpolator for resampling. Bspline, linear and nearest neighbor interpolator are the options available.

6.4.3 Extrusion of vessel

Extrusion of vessel is an important capability for modeling geometry for CFD analysis. As mentioned in section 2.2.3 and 2.2.5, in order to avoid end effects in the flow domain, it is desirable to impose boundary conditions few diameters upstream or downstream from the region of interest which could necessitate generation of cylindrical or non-cylindrical ‘flow extensions’ (Luca Antiga and Steinman 2006). This is particularly true if a geometry is being modeled is tortuous. Recirculation formed as a characteristic of tortuous geometry keeps flow solver from obtaining a

converged solution. Hence extruding the ends of the complicated geometries like patient specific arteries aids in convergence. In this toolkit there are two methods for extrusion of the object of interest.

6.4.3.1 Extrusion along normal/specified direction

The capability of extruding the object of interest is coded in class `vtkKWMimxImageExtrusionGroup` and the flowchart is given in Figure 24. In order to extrude cross section of a segmented object along a normal or any specified direction, a capability of extracting oblique section is necessary as discussed in section 6.4.2. After the 2-D image is extracted utilizing the input center and normal information the extracted 2-D image is prepared for extrusion. Since the algorithm is expecting a binary image at this stage, the default interpolation is nearest-neighbor. Since the geometries being processed are tortuous (e.g. Aorta, coronary arteries etc.), it is possible that a portion from an unintended part of the geometry falls in the extracted plane. Hence, `itkConnectedComponentImageFilter` is used to check how many distinct objects are present in the extracted plane. `itkLabelShapeKeepImageFilter` is then used to keep the object with the maximum or minimum number of voxels as selected by user. The extracted 2-D section is now extruded normal to the plane using `itkZeroFluxNeumannPadImageFilter`. The length of the extrusion is specified by user as a multiple of diameter of the cross section being extracted. As the name suggests the pixels in the 3rd dimension are filled according to zero flux Neumann boundary condition i.e. the first upwind derivatives on the boundaries are zero. The extruded image is then resampled back to the point from where the oblique section was extracted. The section of the segmented image beyond the oblique plane is removed from the original image. Intensities of the two images are then added together to obtain an image with an oblique section of interest extruded in a specified direction.

There are 3 user inputs to this filter: first the center of the oblique plane, second the center that defines the direction of extrusion and third the choice of the cross section to keep (cross section with the maximum or minimum voxels). A unit vector passing along the direction defined by the

two centers/points is calculated and supplied as the direction of extrusion. This method is only applicable in cases when object of interest lies well within the boundaries of the image. If the cross-section to be extruded lies on the image boundary then it may be extruded along the axes as explained below.

6.4.3.2 Extrusion along axes

If the cross-section to be extruded lies on the image boundary then it may be extruded along the axes using `itkZeroFluxNeumannPadImageFilter`. The length of the extrusion is specified by user as a multiple of diameter of the cross section being extracted. As the name suggests the pixels in the dimension of extrusion are filled according to zero flux Neumann boundary condition i.e. the first upwind derivatives on the boundaries are zero. The size of the image increases as a result of this operation.

6.4.4 Remove entities from object of interest

Image erosion or/and opening operators are often used when removing isolated foreground pixels/voxels at edges and that form secluded speckles. Sometimes voxels belonging to unwanted features like small vessels are even painted to background intensity to seclude or obtain the region of interest (Nanduri, Pino-Romainville, and Celik 2009). Both of these methods are supplied in the toolkit.

6.4.4.1 Image erosion

Erosion of foreground pixels is a widely used to remove small entities like small vessels, points of connection between bone and vessels etc. in medical images (Krissian et al. 2014). This is a morphological filter that works by eroding the voxels forming the foreground at the interface of the foreground and background. This capability is available for both grayscale image as well as binary image. User is tasked with the choice of the number of boundary voxels to be eroded.

6.4.4.2 Paint entities within box/sphere widget

`vtkKWMimxPaintImageGroup` and `vtkKWMimxImageErasePixelGroup` are two classes that are built for changing the values of selected foreground pixels to the value of background pixel. In both these classes user selects the region of interest (ROI) in an image interactively. User also has the ability of changing the size of the ROI. Class `vtkKWMimxPaintImageGroup` utilizes box widget to select the ROI such that every voxel that lies inside the box widget is assigned the value of background. Box widget however cannot have arbitrary orientation in space, or in other words it should be aligned with the image axes. This limitation exists because the `RegionType` that can be defined in current version of ITK must be aligned with image axes. User is allowed to select one rectangular ROI at a time and may repeat the process as user finds necessary. The flowchart for this group has been shown in Figure 20

Multiple ROI selection is allowed in class `vtkKWMimxImageErasePixelGroup`. The ROI termed as Landmark are spherical in shape and user has the ability to vary their radius. The spherical widget is useful in situations when the object of interest is tortuous and voxels to converted and localized in small regions. When these landmarks are initialized, Ctrl + left mouse click operation allows user to place a seed at the selected location in the image. Users can place multiple landmarks by same mouse operation. In order to remove any landmark, user can press Ctrl + right mouse button on the landmark. Similarly a landmark grabbed and moved by pressing Ctrl + middle mouse button. This class utilizes fastmarching method to select ROI. The flowchart for this method is given in Figure 21.

When a user places a landmark of a certain radius, say 5 units, on an image, the center of the landmark and the radius are recorded. A blank image of the same size as the input image is created, say image B, and in this image seeds initialized at all the recorded landmark positions. These seeds are given speed of one voxel and are allowed to propagate up to the distance specified by the radius such that the center of the sphere has value of one and the voxels defining the surface of the sphere have value of five. A mask is created such that all the voxels in image B with intensities equal to 0 are assigned intensity of 1 and the remaining voxels are assigned zero

intensity. This mask is then applied to the original image. This capability is useful when one needs to get rid of small connections joining the object of interest with other objects in image. For instance, small connections between aorta and ribcage often have same intensity as aorta and hence leads to segmentation of entire rib cage.

6.4.4.3 Clipping vessel in normal/specified direction

Clipping of the entities in normal or specified direction utilizes the most of the algorithm utilized by class `vtkKWMimxImageExtrusionGroup` for extruding object of interest in normal/specified direction as explained in section 6.4.2. The center and direction information provided by the user is used to extract the corresponding oblique plane from the image. The region in that lies in positive normal direction to the oblique plane is then blanked or given background voxel value.

6.5 Conclusions on manipulation capabilities

In this chapter, we discussed about the capabilities that were added in this work to enable manipulation of the segmented object of interest. This is the first work where all the capabilities for the manipulations of the object of interest are in image domain. Users are expected to perform these manipulations before generating the final levelset.

CHAPTER 7. EXAMPLES OF APPLICATIONS - HEMODYNAMICS STUDIES

Computational based models provide a means of examining flow dynamics by providing the ability to theoretically model and study any possible geometries (Castro, Putman, and Cebal 2006). The main incentive behind building this toolkit is to develop and bring together the capabilities that will allow us to create permutations of patient specific geometries so that we can investigate the flow dynamics in different diseased conditions. There are many biological flows that we are interested in studying but could not pursue earlier due to the limitation of proper segmentation and manipulation capabilities. With the advent of this toolkit and our in-house flow solver those studies are straight forward. In this chapter we will talk about examples of some challenging questions we can now easily ask using the capabilities developed in this work.

7.1 Effect of the orientation of flow extension

In order to avoid the end effects in hemodynamic simulations, it is desirable to impose boundary conditions few diameters upstream or downstream from the region of interest. For this purpose, ‘flow extensions’ are commonly added at inlets and outlets when preparing models for hemodynamic simulation (Juan R. Cebal et al. 2004; Luca Antiga and Steinman 2006). As mentioned in sections 2.2.3 and 6.4.3, this feature is particularly important when doing CFD analysis in tortuous geometries. While it is common to prepare flow extensions normal to the cross section at the inlets and outlets, our group did not have this capability in our image to flow setup. Hence in our previous works (S. I. Dillard et al. 2014) we have used flow extensions parallel to the axes. As explained in section 6.4.3, we have added the capability of generating flow extensions normal to the inlets and outlets boundaries in this work. To understand the benefit of this capability a study was devised to investigate the effect of orientation of flow extensions in the region of interest.

Two models were created after conducting preprocessing segmentation of ICA. In the first model - ‘Model 1’, the ends of the vessels were cut by the planes parallel to the axes and flow extensions were created normal to the obtained cross sections utilizing

VtkMimxCalculateCenterGroup and VtkMimxNormalExtrusionGroup. A second model – ‘Model 2’ was created by cutting the cross sections at the inlet and outlet normal to the geometry and extruding the cross section. A comparison of the two models of ICA so created are presented in Figure 28, where the Model 1 is colored in white and Model 2 in red. As mentioned earlier, we only had the ability to create Model 1 before this work. Figure 29 and Figure 30 show results of flow simulation in Model 1 and 2 where pressure and velocity vectors are plotted. The pressure distribution in Model 1 is higher at the inlets and lower at the outlets than Model 2. Analysis of the plots of cross sections colored by pressure in Figure 31 clearly shows that velocity vectors entering the ICA from inlet 1 for Model 1 have magnitude higher than Model 2 by more than 0.4 m/s. This is true for pressure distribution too.

Investigation of the cross section at the inlet of the aneurysms as shown in the Figure 29 demonstrates the difference in flow waveform at the entrance of aneurysm. Magnitude of velocity vectors were 4.4 m/s and 3.7 m/s for Model 1 and 2 respectively. Hence, we found that the orientation of the inlet extensions does effect the hemodynamics in the region of interest. We must note that before this work, we did not have the capability of creating flow extensions perpendicular to the inlet or outlet in our image to flow setup. Addition of this capability has enabled us to model patient specific geometries more accurately.

7.2 Analysis of flow features in the diseased vessel with aneurysm and without aneurysm

One of the applications of this toolkit could be preparing an approximation of a healthy model of a vessel if we only have images of vessel already diseased with aneurysm. There are many approaches one can take to create an approximation of a healthy vessel. For example, previously in a CFD study done by Cebal (2011), they studied the reason of formation of bleb, a mechanobiological damage on the wall of aneurysm caused by inflow jet, in cerebral aneurysm. They created models with and without bleb for each aneurysm they studied. In the later cases, the bleb was removed by smoothing vascular model obtained from the 3DRA image (J. Cebal et al. 2011). Similarly, Singh (2010) used @neuFuse software to remove the aneurysm and studied WSS

and OSI in the ‘pre-aneurysmal’ vessels. They varied the viscosity of blood to mimic the alteration in blood due to smoking and conducted CFD analysis.

We obtained the CTA image of the vessel with saccular or sideways aneurysm. A saccular aneurysm is bulge in vessel where the ostium of the bulge is entirely contained in a part of the circumference of the wall of the vessel (P. C. Karmonik and Britz 2014). We noted that the maximum diameter, height of this aneurysm are small and its neck is broad when compared to the parent vessel indicating that the parameters like bottleneck factor and aspect ratio of the aneurysm are small and thus the vessel should not rupture (Hoh et al. 2007). This observation is supported by the fact that in a follow-up scan done after a year, the size of the aneurysm was found constant. Though we knew that the aneurysm would not rupture we thought it was still worth investigating the difference in hemodynamics in a vessel with aneurysm and without aneurysm. Since we only had image of vessel with aneurysm, we decided to remove the aneurysm bulge from the segmented image of the diseased vessel for modeling of healthy vessel (Singh et al. 2010; J. Cebal et al. 2011). In this diseased vessel segment, aneurysm is present in a tortuous region of the vessel followed by a bifurcation. Segmentation of the diseased vessel was performed using general segmentation scheme (Method I) explained in section 4.3. An approximation of a healthy vessel was then prepared by removing the voxels forming the saccular aneurysm by applying `vtkMimxEraseImagePixelGroup` and `vtkMimxPaintImageGroup`. Both these models were subjected to the same boundary conditions. Patient specific velocity was prescribed at the inlet and the Reynolds number was 248, and the outlets were assigned mass split of 50:50.

Rupture of an aneurysm is often linked with a formation of a jet hitting a localized region of aneurysm wall (J. Cebal et al. 2011) hence we looked for the formation of jet and localized region of high pressure in the diseased vessel in comparison to the healthy vessel. A comparison of pressure gradient (Figure 32) showed that pressure in the diseased vessel was in average higher than in the non-diseased vessel by about 12.5% which is only about 5 Pa. A closer look at the pressure distribution in the slice cut in the middle of aneurysm in the diseased case (see inset (a) in top image in Figure 33) showed the highest pressure along the wall of the aneurysm and a least

towards the center. The pressure distribution looks qualitatively similar for the corresponding slice in the healthy case (see bottom image in Figure 33), only the magnitude is different. Pressure along the outer wall of the diseased vessel is higher than the healthy vessel by 1 Pa. Similarly, the pressure in the center region in the slice is higher in the healthy vessel by about 1 Pa. Figure 33 clearly shows that in the presence of an aneurysm, regions of high and low pressure exist where pressure was previously more uniform, and a new flow recirculation region can be observed where secondary flow was not present in the healthy vessel (See insets (b) and (c) in Figure 33. We did not observe any jet formation or stream of blood hitting any localized region of aneurysm, which are linked with aneurysmal rupture (J. Cebal et al. 2011). This observation is attributed to the fact that the maximum diameter, height of this aneurysm is small and its neck is broad when compared to the parent vessel. Hence, our observations are in line of the findings that rupture risk in aneurysm is proportional to aspect ratio and bottleneck factor.

7.3 Effect of root on hemodynamics and study of rupture risk in ascending aorta

Aortic rupture results from mechanical failure of the aortic wall, occurring when the acting wall stresses exceed the tensile strength of the tissue (Vorp et al. 2003). In aortic dissections, the variation of aortic wall material properties, while also important, has not been modeled in the current simulation, to isolate the effects of flow characteristics impacting the aortic wall, and by extension, regional rupture risk. In fact, in the context of aortic dissection, with current technology, it is unrealistic to predict the regional effect of the dissection on aortic wall resistance as a consequence of the variability of the pattern of dissection on a case-by-case basis. Nevertheless, the dissected aorta seems to be more susceptible to rupture in certain areas compared to others and the distribution of wall stress can be calculated and predicted based on the study of the aortic flow dynamics and the aortic geometry. In fact, following the basic principle of Aristotle scientific method, according to which knowledge can only be gained by building upon what is already known, we have based our study hypothesis on the knowledge that the dissected aorta in Type-A dissection ruptures in the vast majority of cases at the aortic root and the proximal ascending aorta,

within the pericardial reflection (Marc R. Moon 2009; M. R. Moon et al. 2001). This suggests that some aortic segments are more susceptible than others to rupture and mechanical failure. Also, reviewing our clinical experience on operated Type-A dissections with arch involvement (DeBakey type-1), it was observed that when aortic resection did not include the arch and descending thoracic aorta, the aortic arch was much less likely to require replacement at a later time, as compared with the descending aorta. This indicates that the potential exists to adopt a clinical strategy that limits surgical replacement only to the ascending aorta in cases of Type-A dissections, even when the arch is involved. For this reason, we aimed to determine if the observed local variation in rupture risk was attributable to regional characteristics of the aortic flow dynamics.

7.3.1 Methods

We employed CFD to examine the relationship between aortic dissection rupture risk and characteristics of blood flow in an idealized aortic model during key phases of the cardiac cycle. To do this, we employed four idealized geometries representing the aortic valve and aortic arch regions, two at peak systole and two in early diastole. We currently do not have the capability of segmenting valves hence we constructed idealized models for valve and sinus for this study. Our model incorporated the dimensions of an average human aorta (Vasava et al. 2012) with normal tricuspid aortic valve (TAV) and coronaries (Kahraman et al. 2006; Dodge et al. 1992) rooting from three Sinuses of Valsalva (Burken 2012; Jermihov et al. 2011). The dimensions of the TAV were similar to the physiological values, with radius equal to 11.5 mm and height of 10.9 mm. The aortic root was designed with dimensions as follows: The inlet diameter was 27.8 mm; the outlet diameter was 25 mm; the diameters of the brachiocephalic (BC), left carotid (LC), left subclavian (LS), left coronary, and right coronary were 8.8 mm, 8.5 mm, 9.9 mm, 4.5 mm and 4 mm respectively. The aortic arch was modeled to have a radius of curvature of 32.5 mm. The descending aorta was modeled as a tapered tube having an outlet of 20 mm diameter 80 mm downstream. The descending aorta is inclined at an angle of 10° with respect to the y-axis (Vasava

et al. 2012). For modeling purposes, the flow inlet at the aortic root was extended by 5 mm to reduce end effects. Below are the three scenarios considered, as seen in Figure 36 (A-C).

1. Case A: Aorta with sinus and valve in fully open position (peak systole)
2. Case B: Aorta without sinuses and valve (peak systole)
3. Case C: Aortic valve in fully closed position with coronary branches (early diastole – maximum reverse flow)

Case C was included to investigate the significance of the aortic valve and sinus geometry on the flow distribution within the ascending aorta and aortic arch region. Previous work that has modeled hemodynamics in this region (Vasava et al. 2012; Tan et al. 2009; Tse et al. 2011; Chen et al. 2013; Karmonik et al. 2013) did not typically include the aortic valve apparatus, and we hypothesized that the inclusion of the Sinuses of Valsalva and aortic valve in the flow simulation would have a significant impact on the aortic arch hemodynamics.

The idealized geometries of the aortic arch, sinus and leaflets were constructed in Rhinoceros (Robert McNeel & Associates, Seattle, WA) and an unstructured tetrahedral mesh was then created in Pointwise (Pointwise, Inc, Fort Worth, TX). The average interval size of the mesh was 0.5 mm. The resulting mesh had around 5 million tetrahedral cells in each vessels geometry. The mesh was then imported into Fluent 12.1.4 (ANSYS, Inc, Canonsburg, PA) to analyze the hemodynamics. The table below summarizes the total number of tetrahedral cells present in each of the idealized cases.

Table 4 List of number of cells in different cases simulated.

	Peak systole with sinus and valve	Early diastole with sinus and valve	Peak systole without sinus and valve
Number of Tetrahedral Cells	5,619,298	5,341,638	4,883,327

Blood was assumed to be a Newtonian fluid with the density of 1050 kg/m-s and a viscosity of 3.5cP. Both during systole as well as diastole, a uniform inlet velocity was assumed. Walls were assumed to be rigid with no slip boundary condition. For peak systole, zero pressure and velocity outlet boundary conditions were specified at the outlet of the descending aorta and head vessels, respectively. Mass split among the head vessels was specified based on cross-sectional area. A mass flux of 20 L/min was prescribed at the inlet for peak systole simulations. For early diastole, coronary arteries were set to have a zero pressure at outlets. Mass influx to the coronaries we assumed to be 100 mL/min each (Chaichana, Sun, and Jewkes 2012; Ramaswamy et al. 2006; Sabbah et al. 1984; Zeng et al. 2008). Proportions of the mass influx into the head vessels with respect to the total mass influx during diastole were assumed to be same as in systole. The convergence was set to $1e^{-04}$ and the scheme was set to SIMPLE first order upwind. The time step size was taken to be 0.01s (Burken 2012) and the simulation was performed until results converged. Wall shear stress, pressure, and velocity were then examined to determine whether their local distribution may explain the observed differences in rupture risk.

7.3.2 Results

We sought to analyze the roles played by fluid pressure and stresses in an idealized human aorta during the key phases of maximum hemodynamic stress in the cardiac cycle: peak systole (with the aortic valve fully open) and early diastole (maximum reverse flow, with aortic valve

closed). Our results are organized based on several hemodynamic measurements, including velocity, wall shear stress, and pressure, in each of the cases simulated.

Velocity distribution at peak of systole

During systole blood passes through the aortic valve forming a jet that impacts the aortic arch. Fluid energy gets dissipated in the sinuses as the high pressure fluid, entering through a comparatively narrow lumen valve, finds a larger area to diffuse and as a result, a recirculation region is observed in the sinuses. These flow characteristics are clearly seen in Figure 37 and Figure 38, where velocity distribution in the coronal plane (Figure 37) and streamlines of velocity magnitude colored by pressure (Figure 38) are plotted. In Case A, we can clearly see the region of maximum velocity (>1.1 m/s) colored in red, and the recirculating flow in the sinuses, in blue. The velocity is highest on the inner wall of the aortic arch, and the jet dissipates as the flow divides between the head vessels and the descending aorta. Recirculating flow (colored in blue (0-0.2 m/s)) in the sinus region, and the velocity jet colored in red (within the inlet and valve regions) are both characteristic of flow passing through the aortic valve (Bissell, Dall'Armellina, and Choudhury 2014; Gharib et al. 2002).

As can be observed, very different flow dynamics exist when the aortic apparatus is modeled without the aortic valve and sinuses. In this scenario, a velocity jet is not established, and no flow recirculation is observed (Case B – top right, Figure 37 and Figure 38). Similarly, the magnitude of velocity remains lower and more uniform throughout the ascending aorta, with the velocity ranging from 0.8-0.9 m/s in Case B, as compared to the flow distribution range from 0.1 -1.13 m/s in Case A, when the aortic valve and sinuses are included in the model.

Pressure distribution at peak of systole

In an effort to determine the nature of the pressure distribution and particularly the locations of peak pressure, we examined the hemodynamic pressures in the model during maximum forward flow (peak systole). As seen in Figure 37 (contours), pressure is highest at the inlet to the valve ($>10,200$ Pa) and in the ascending aorta ($>10,100$ Pa). It gradually decreases in magnitude within the arch region, as a fraction of fluid diverts through the head vessels ($\sim 10,050$

Pa) while the remaining flow passes into the descending aorta (<9950 Pa). It is clear that the head vessels in the arch region are reducing pressure from the aorta by splitting the fluid mass from the aorta to the head, neck, shoulders, and lower extremities of our body. Similarly, the sinuses, due to their larger area, also help to dissipate energy generated at the valve inlet. The aortic valve region is seen to have pressure higher than 9440 Pa at the core, while in the sinuses, the average pressure is found to be lower by at least 100 Pa (Figure 37 see inset). The pressure distribution remains more uniform, and peak pressure is substantially lower, when the aortic valve and sinuses are not incorporated into the geometry, as shown in Figure 37(Case B - top right). On investigating the pressure distribution in the sinuses, contours of pressure at peak of systole (shown in inset in Figure 37 (Case A)) pressure was highest in the non-coronary sinus compared to the left and right coronary sinuses.

Velocity and pressure distribution in early diastole

For diastole, in order to reproduce the reverse flow that occurs when the valves are closed, the descending aorta has been treated as velocity inlet while head vessels and coronary arteries have been treated as outlets. As blood from the descending aorta moves towards the ascending aorta during early diastole, pressure gradually decreases (Figure 38, bottom) as it approaches the head vessels (from > 9370 Pa to < 9369 Pa). Pressure further decreases after the head vessels though the difference is lower than 1 Pa. This reinforces our hypothesis that the head vessels contribute to relieve pressure. On analyzing the pressure in the sinus region as shown in Figure 37 (inset (Case C)), the non-coronary sinus clearly has higher pressure than the left and right sinuses. This is primarily due to the fact that non coronary sinus lies in the outer wall of the aorta where velocity of fluid is higher than the inner wall of aorta, due to centripetal force. Secondly, as the name suggests, the non-coronary sinus is devoid of any coronary artery and thus fluid entering this sinus does not get a direct outlet, leading to higher pressure. So the pressure is higher in the non-coronary sinus throughout the cardiac cycle. This finding is in line with the fact that most dissections happen in the non-coronary sinus (Komiya et al. 2009), and it could be due to the higher pressure in this region throughout the cardiac cycle. This occurs because the non-coronary sinus

is anatomically the largest, and is subject to the strongest hemodynamic stress because of the characteristics of the flow dynamics within the aortic root (Grande et al. 1998).

Wall shear stress

Figure 38 shows contours of wall shear stress obtained in case A, case B and case C and arranged as insets at top left, top right and bottom respectively. WSS during diastole is negligible in comparison WSS during systole, as the range varies by two orders of magnitude. During systole, WSS is highest in the regions where jet impinges the aorta (the leaflets and inner wall in this context). The sinuses have very low WSS throughout the cardiac cycle because of lower fluid velocity.

7.3.3 Discussions

After simulating the flow in a computational model of an idealized human aorta with and without sinuses and valve (Figure 36), we measured the regional variation of pressure distribution, wall shear stress and velocity. Our selected models captured peak systole and early diastole to simulate the times in the cardiac cycle where maximum forward and reverse flows exist (Chandran, Rittgers, and Yoganathan 2007). We observed that for all models, flow is initially directed towards the inner wall and as it approaches the arch, flow becomes skewed towards the outer wall (Figure 38). Skewness is more pronounced in Case B which is devoid of a valve and hence lacks the characteristic jet flow produced by the aortic valve. The pressures at the inlet, valve region, and ascending aorta are clearly higher compared to the pressure within the arch both at peak systole and in early diastole (Figure 37). From this it can be hypothesized that the presence of the great vessels relieves some of the wall stress that would be otherwise induced by the impact of flow, the so-called “head loss.”(Chandran, Rittgers, and Yoganathan 2007) Additionally, the pressure distribution measured on the cross-section of the aortic root shows higher pressure in the non-coronary sinus during both systole and diastole, which supports the remodeling theories of asymmetric stress distribution on the aortic valve leaflets and the walls of the root sinuses (Grande et al. 1998), and explains the higher incidence of development of aortic dissection in the non-

coronary sinus in compare to the rest of the aortic root. Similarly, WSS plots (Figure 39) suggest that the inner wall of aorta is subjected to highly fluctuating WSS as it receives higher WSS in the direction of flow during systole and negligible WSS during diastole, which is intuitive as WSS increases with increasing velocity (Chandran, Rittgers, and Yoganathan 2007).

Our observations suggest that during systole, blood pressure is significantly higher in the ascending aorta than in the arch or descending aorta while during diastole, the arch region experiences lower pressure than the descending aorta. This can in part influence the regional rupture risk of a dissected aorta, and supports the greater need to replace the ascending aorta than the arch region.

7.3.4 Conclusions

Our computational models of aortic fluid dynamics showed higher pressure, wall shear stress and velocity magnitude in the root and ascending aorta compared to the arch in both systole and diastole. This suggests less wall stress in the arch compared to other aortic segments supporting the clinical evidence showing that risk of rupture after Type-A dissection is greater in the root and ascending aorta. To our knowledge, this is the first study of computational fluid dynamics of the root, ascending aorta, and arch, defining the characteristics of wall stress during the key phases of the cardiac cycle in correlation with the possible effects on aortic dissection. The application of the results of this investigation to clinical practice may help in understanding the distribution of the regional rupture risk in the dissected aorta, possibly guiding the principles of surgical repair. In our model, the inclusion of the sinuses and valve has allowed to create conditions of simulation which were pivotal in capturing realistic flow profiles.

7.4 Understanding hemodynamics in dissected aorta

7.4.1 Introduction and Motivation

An aortic dissection is a condition created by tears in the intimal layer of the aorta which causes the formation of a false lumen for blood flow. Aortic dissections have been classified using

the Stanford and the DeBakey classifications. The Stanford classification is based on whether there is involvement of the ascending aorta, whereas the DeBakey classification is focused on grading the extent of the dissection throughout the vessel (DeBakey et al. 1966; Daily et al. 1970). In Stanford classification, dissection can be categorized in two major groups depending on the location of the dissection septum with respect to the ascending aorta and heart. If the false lumen exists in the ascending aorta and arch region, the dissection is categorized as a Stanford Type-A dissection. Instead if the false lumen exists only in the descending aorta then the dissection is categorized as Stanford Type-B dissection. Type-B dissections can be treated medically while Type-A dissections require immediate surgical measures. Propagation of false lumen leads to the obstruction of branching vessels both proximally and distally within the true lumen of the dissection, thus causing insufficient blood flow in those areas. Type-A dissection is considered more dangerous because of its direct impact on blood flow to the heart and brain, and the fact that the propagation of the false lumen towards the heart can lead to life threatening complications like cardiac tamponade, coronary obstruction, and aortic insufficiency (Hirst, Johns, and Kime 1958). Hence, Type-A dissections (DeBakey 1 and 2, Figure 14) require immediate intervention, and are associated with a much higher risk of death (50% in the first 48 hours). Type-B dissection, on the other hand, which are 50% less common than Type-A, are associated with much lower mortality and require surgical intervention less frequently (Coady et al. 1999; Urbanski et al. 2010). A major dilemma for cardiac surgeons in the management of Type-A dissections is determining the extent of surgical aortic replacement required in cases where a Type-A dissection is associated with involvement of the arch (DeBakey Type 1, Figure 14) (M. R. Moon et al. 2001). Is the replacement of the arch required in all cases of Type-A dissection involving the arch? On this subject there is a large controversy amongst aortic surgeons as to whether to replace the aortic arch at all times, or limit the aortic replacement only to the ascending aorta (Moon et al. 2001; Urbanski et al. 2010; Geirsson 2010; Bachet et al. 1996). The problem is substantial because including arch replacement significantly increases the morbidity and mortality of the operation (M. R. Moon et al. 2001; Marc

R. Moon and Miller 1999; Crawford et al. 1992; Borst, Bühner, and Jurmann 1994; Haverich et al. 1985).

In the review of a 7-year experience of Type-A dissections performed at the University of Iowa between 2004 and 2011, it was found that among a total of 74 patients, 62 had arch involvement (DeBakey Type-1). Four of these patients underwent total arch replacement and 9 expired; three patients were lost at follow up. We were left with a group of 46 patients with aortic dissection and arch involvement who had undergone surgical repair without replacement of the arch. This group underwent serial surveillance with computed tomography (CT) at regular time intervals (1 month, 6 months, 1 year, and yearly afterwards). Most patients were dismissed from serial CT-scan surveillance after 5 years if no significant changes compared to previous exams were identified. At a mean follow up of 28 months (6-60), only one patient required arch replacement, compared to 4 who required operation for replacement of the thoracic or thoraco-abdominal aorta.

Based on this study, we decided to look into hemodynamics in Type-A aortic dissection by employing CFD. We used patient specific data and carried out pulsatile flow simulations to better understand the effect of Type-A aortic dissection with respect to a healthy aorta, location of primary tear and secondary tear; and the impact of repair without the replacement of arch. Previously researchers have investigated hemodynamics in Type-B aortic dissections, however, there are no hemodynamic studies on Type-A dissection to the best of our knowledge.

7.4.2 Previous studies in literature

A number of groups (Christof Karmonik et al. 2013; Christof Karmonik et al. 2012; Alimohammadi et al. 2016; Chen et al. 2013; Tse et al. 2011) have looked into patient-specific hemodynamics in the dissected aorta using computational fluid dynamics to study Type-B dissection.

For example in two consecutive studies done in the same geometry with Type-B dissection (C. Karmonik et al. 2010; C. Karmonik et al. 2011), Karmonik et al. in 2010 studied the impact of

occlusion of entry and exit tear individually. They found that occlusion of entry tear as an endovascular treatment (EVAR) would result in a decrease in flow in false lumen while occlusion of exit tear as in thrombus formation would lead to high pressure which could lead to rupture. In another study in 2011, Karmonik et al. compared the hemodynamics in dissected and normal aorta where the normal aorta was created by merging the true lumen with false lumen and dissection. They noted lower velocity and pressure at the entrance of the tear in the normal case when compared with the diseased case. Another CFD study on pre- and post-stent placement (EVAR treatment) in a patient with Type-B aortic dissection was done by Karmonik et al. (2011). They found a reduction in WSS and dynamic pressure in the post-treatment case in compare to pre-stent case suggesting reduction in risk of rupture. However, they noted two specific areas of high dynamic pressure within the stent graft and which they suggested could lead to stent migration (Christof Karmonik et al. 2011).

In 2011, Tse et al. investigated the influence of aortic dissection in the development of aneurysm in the false lumen of the dissected aorta with Type-B dissection (Tse et al. 2011). They used images obtained pre and post aneurysm development in an aorta whose ascending aorta was already repaired as a treatment of Type-A dissection. They segmented the images in Mimics v13.0 (Materialise, Leuven, Belgium) and prepared tetrahedral mesh using HyperMesh (Altair HyperWorks, Troy, MI, USA) and performed pulsatile flow simulations in ADINA v8.6.0 (ADINA R&D, Inc., Watertown, MA, USA). They found that high pressure difference of 0.21 kPa between true lumen and false lumen in pre-aneurysmal aorta may infer to the possible lumen expansion and development of aneurysm.

Karmonik et al. (2012), in a similar study on the effect of aneurysm dilation on patient with type III DeBakey dissection, examined the difference in hemodynamics of at two different stages of aneurysm progression. For a dilation in cross section by 26% they found a 29% decrease in total pressure during systole and 34% decrease during retrograde flow. Similarly, they also noted a decrease in maximal average WSS. They put forward a hypothesis that dilation of false lumen in chronic DeBakey type III dissection may be a compensation mechanism to decrease the force

directly exerted on luminal wall. In a consecutive study, Karmonik et al. compared the hemodynamics in the previously studied dissected model with aneurismal dilation against hemodynamics analysis of a different healthy patient (Christof Karmonik et al. 2013). They compared the results with the data obtained from 2d pc MRI. They reported lower average WSS and pressure in peak systole and retrograde flow in healthy aorta in compared to the dissected aorta.

Chen et al. 2013 looked into flow exchange between true and false lumen through entry tears in a Type-B dissection model which consisted of a primary, exit and a secondary tear. He segmented the model using region growing based segmentation using ShIRT toolkit, modeled the geometry in ICEM CFD and performed CFD simulations CFD-ACE+ and ESI CFD. He supplied pulsatile waveform at inlet and assigned 5% flow at each head vessels. In his investigation he found that about 40% of total flow was diverted to false lumen in a cardiac cycle for this patient. He also found comparable results when using fine mesh Vs a coarser mesh with only refinement in boundary elements.

A study was performed in simpler model by Alimohammadi et al. 2014; they incorporated coupled RCR Windkessel models at the outlet branches for dissected geometries to incorporate more realistic boundary conditions mimicking the resistance offered by distal vessels. In this study, the Windkessel parameters were tuned using invasive pressure measurements. Their results showed that only 45% of the total blood entering the aorta passed through the descending aorta accounting for severe risk for limb and/or vital organ malperfusion syndrome which is generally the case for patient with aortic dissection.

A thorough investigation of the effect of size and location of tear in 25 patients with Type-B dissection was performed by Rinaudo et al. 2014. They used semi-automated thresholding from VMTK to segment the ascending aorta, arch and head vessels; and 3D aortography to model multiple re-entry. The models were meshed in GAMBIT 2.3.6 (Ansys Inc., Canonsburg, PA) and flow was simulated in FLUENT v15.0.0 (ANSYS Inc., Canonsburg, PA). They supplied a pulsatile waveform at the ascending aorta such that a cardiac output of 5L/min was achieved. Inlet flow

volume was then split between head vessels in a ratio of 10% at brachio-cephalic trunk, 4% at left common carotid, and 6% at left subclavian artery and 70% at the descending aorta. They computed flow and determined size, particularly the height of tear, and proximal location of the entry tear to be predictors of blood flow through false lumen and Type-B dissection related adverse effects.

Lastly, Dillon-Murphy et al. 2015 performed hemodynamic simulations in a patient specific Type-B dissection and validated the results against 4D PC-MRI data. They included all the branch vessels throughout the aorta in their model and applied flow waveforms obtained through 2D PC-MRI data. They looked at the difference in hemodynamics in healthy Vs dissected model and found an elevation in pressure, WSS and velocity in the region of primary tear in dissected aorta in compare to the healthy one. They compared of quantity of blood flow through the false lumen and true lumen in a cardiac cycle between two models; one comprising of tiny connection tears and the other devoid of those tears. They found that hemodynamics in the dissected aorta was sensitive to the modeling of secondary tears that are too small to image.

Importantly, in all of the above mentioned work, researchers have examined the hemodynamics of Type-B dissection. From these studies we have learned about the importance of location and size of primary tear, size of the true lumen, and contribution of secondary tears on flow in the true lumen and distal branches. To the best of our knowledge, this is the first study done that focuses on the hemodynamics of Type-A dissection. The capabilities developed in this dissertation, and specifically those described in Chapter 6, have allowed the modeling described below to be performed in a straightforward fashion. In this way, segmentation of the patient-specific Type-A dissections, and manipulations to the tears in dissection, were possible with relative ease. In this work, the alterations in hemodynamics resulting from changes in the tearing profile, location, and false lumen size in Type-A aortic dissections have been investigated and are described below.

7.4.3 Material and methods

In this methods segment we will talk about the image used, segmentation method, object manipulation methods, boundary conditions, grid independence study, and CFD.

7.4.3.1 Imaging data

A CT scan of a patient with Stanford Type-A dissection comprising of 211 axial images with in-place resolution 512 x 512 was used for this study. The images in this dataset have spatial resolution of 0.822 x 0.822 mm and inter slice distance of 3 mm. The subject had an acute Type-A dissection originating at the junction of ascending aorta and arch and extending the full length of descending aorta and terminating few mm proximal to the iliac bifurcation. The dissection and comprised of 4 subsequent tears, one towards the middle of descending aorta (location (i) in Figure 46 (c)) and three towards the end of descending aorta (locations (ii) & (iii) in Figure 46 (c)). Head vessels and right renal artery branched out from the true lumen while left renal artery branched from the false lumen. It should be noted that left subclavian artery appeared to be partly blocked or ‘pinched’ due to the false lumen.

7.4.3.2 Geometry modeling

Most techniques give satisfactory result for segmenting vascular structures if the underlying image quality is good. However it is difficult to obtain good quality segmentation due to complication in the geometry because of varying size and shape, tortuosity and curvature, bifurcations and variation in intensities. These complications become more severe in diseased vessels. Segmentation methods applied for dissection in literature are region growing (Chen 2013), intensity-based region-growing manual inspection and correction (Shang 2015), active contour segmentation (Krissian 2014), Hough transform (Cattian 2006).

Semi-automated segmentation methods proposed by Krissian et al have been used for MDCT images that do not have drastically different intensities in true and false lumens. In addition to that they have only used Type-B dissections and post operation Type-A dissection patient images. Further, the main objective of the work is extraction of dissection wall. They have not

performed hemodynamic simulations. Region growing is based on centerline computation followed by the calculation of local mean and variance, to create an expansion image. This image is subjected to geodesic active contour segmentation where curvature value is increased to include dissection wall in the outer lumen segmentation. Resulting image is multiplied with denoised image. This enables easy segmentation of the dissection wall as it has non-zero intensity but lower than the lumen intensity. This is followed by determination of center sheet representation of wall using Eigen values. In most CFD simulations performed on dissection commercial software like Amira, VMTK (Rinaudo et al. 2014) have been used for segmentation and 3d reconstruction.

In this work images of patient specific Type-A dissections have been segmented and smooth 3D model for CFD simulation was created following Method III as explained in 4.5. which uses techniques incorporated into this platform that are based on region growing, statistical analysis and geodesic active contours. The model was then meshed using Pointwise. The details of segmentation method and validation can be found in 4.4 and 4.5. Several permutations of this model was created by adding and removing connecting tears between the true and the false lumen using paint pixel algorithm explained earlier in section 6.4.4.2. Flow extensions were created at the inlets and outlets using extrusion algorithms from section 6.4.3.1 and 6.4.3.2 respectively.

7.4.3.3 Grid independence study

We carried out test simulations for peak systole case on current models of Type-A dissection in our in-house flow solver. However, the size discrepancy between the collapsed true lumen and ascending aorta led to huge difference in Reynolds number. Our flow solver utilizes adaptive grid refinement based on velocity, and for such discrepancy in Reynolds number will require allocation of huge number of nodes and memory. We were not able to achieve the level of refinement required with the current computational capacity hence we have requested for more computing capacity to carry out this simulation in future. Hence, for the purpose of this thesis we decided to perform flow simulations in FLUENT.

A grid independence study was performed on the patient-specific Type-A aortic dissection in order to verify that the results of the simulation are independent of the mesh employed. For this study, the segmented model was meshed using three different average mesh size: 0.002 mm, 0.001 mm and 0.00075 mm and they are described as coarse, medium and refine models. The coarse, medium, and fine meshes resulted in 100,310, 245,689, and 427,533 nodes and 478,692, 1,278,447 and 2,284,620 tetrahedral elements, respectively. A transient simulation was carried out with steady flow at inlets and outlets to simulate peak systole case. An inlet velocity of 0.6167 m/s was prescribed at ascending aorta and outlet velocities of 1.162 m/s 0.2924 m/s, 0.55 m/s 0.52 m/s and 0.035 m/s were prescribed at the descending aorta, brachio-cephalic1, brachio-cephalic2 left common carotid and left subclavian arteries respectively. Blood was assumed to be a Newtonian fluid with density and viscosity of 1060 Kg/m³ and 0.0035 Poise. Simulations were run with residual control of 1×10^{-3} . Six random cross sections as shown in Figure 58 were analyzed for velocity vectors and average velocity magnitude was computed. Calculations suggested that medium and coarse mesh models have relative average errors of 4.06 and 9.66 percent when compared with the finest model. Medium case with average element size of 0.001 mm was selected for this study.

7.4.3.4 CFD studies

Patient specific pulsatile flow waveforms for different inlets and outlets as shown in an inset in Figure 44 was supplied as boundary condition at the inlets and outlets. We did not have the patient specific flow data of the subject, hence we employed flow waveforms of another patient for this study. Flow simulations were done in FLUENT v14.5.0 (ANSYS Inc., Canonsburg, PA) Vessel wall was assumed to be rigid (Ganten et al. 2009; Chen et al. 2013b), blood was used as a Newtonian fluid and pulsatile simulations were run for 3 cardiac cycles with residual control of 1×10^{-3} . Cycle to cycle periodic solution for pressure and flow was checked for convergence. In this work, the hemodynamic alterations between different models of Type-A aortic dissections have been investigated by performing these studies.

An investigation of alteration in hemodynamics introduced by dissection between a diseased model with Type-A dissection and a baseline ‘healthy’ aortic model is studied. The baseline healthy aortic model is created by merging the true lumen, false lumen and dissection septum found in the original dissected geometry which is termed as ‘Diseased’ model in this study.

A quantification of the impact of repair done on Type-A dissection model by patching the primary tear. This study helps to quantify the amount of mass flux restored in true lumen as a result of the repair and understand the impact of secondary tears that still exist on the repaired models. In this study three different dissection models were studied by varying the number of tears.

A quantification of mass flux in the false lumen and pressure in the primary tear region if the location of the primary tear is altered in Type-A dissection. This study also gives an insight on the success of repair surgery with respect to the secondary tears still existing in the aorta. In this model four different models were investigated where the location of the primary tear was altered for the same dissection.

A sensitivity study of the impact of modeling secondary tears in Type-A dissection. Aortic dissections comprise of multiple tears throughout the aorta. However, these tears may not get captured properly during 3D modeling due to noise, motion artifacts, low spatial resolution of images, manual judgement in segmentation etc. In current diseased model we came across a secondary tear that could be easily missed during segmentation. Hence we created two instances of Diseased models by including and excluding the secondary tear.

7.4.4 Results

7.4.4.1 Study 1: Hemodynamic alterations in aortic dissection

The first study using the patient-specific Type-A aortic dissection geometry was performed to understand the effect of a dissection on the hemodynamics in an aorta. For this purpose, two models were considered: the first is the original dissected or “Diseased Model” obtained from a CT scan of a patient, and the second, a “Baseline Model” representing a healthy aorta which was prepared by merging true lumen, false lumen and dissection from the Diseased Model, using an

approach previously performed in a Type B dissection to similarly replicate a healthy/baseline geometry from a dissected geometry (Christof Karmonik et al. 2012; Dillon-Murphy et al. 2015). These models are shown in Figure 43. An analysis of pressure in these two cases show that dissection causes rise in pressure in the ascending aorta and the arch region. In comparison to the baseline undissected model, the presence of dissection has reduced the amount of blood flow in the true lumen by an average of 30 % in a cardiac cycle and increased pressure in ascending aorta and arch by 12 mmHg in peak systole as shown in Figure 44. This is in agreement with the findings of other research groups like Tse et al. (2011); Dillon-Murphy et al. (2015); Alimohammadi et al. (2016) who have noted an increase in pressure in the region proximal to the primary tear. Pulse pressure (pressure at peak systole/ pressure at early diastole), similarly has increased from 28/12 mmHg to 40/20 mmHg. A closer look at the region adjacent to the primary tear shows that the maximum and average pressure in the tear was 38 mmHg and 6.7 mmHg. The area where the jet from the true lumen impinges on the wall of false lumen (see insets in Figure 44) displays maximal pressure throughout systole. Such a localized region of high blood pressure in an already structurally compromised false lumen could lead to enlargement and potential rupture (Dillon-Murphy et al. 2015). A comparison of Time-Averaged Wall Shear Stress (AWSS) between the two models shows that dissected model experiences higher AWSS throughout the aorta (See Figure 45). The maximum spike in AWSS from 5 N to 16 N were seen at the region of entry tear and from 5 N to 10-16 N in the region of minimal cross section in the true lumen.

These results indicate that an aortic dissection causes an increase in average pressure, and velocity magnitude in the true lumen, and significantly reduces the amount of flow in true lumen. These observations have important implications to long-term survival after repair, as these flow alterations could lead to insufficiency in blood flow in the organs downstream.

7.4.4.2 Study 2: The impact of repair on Type-A dissection

The standard of care for Type-A dissection calls for surgeons to remove the entire diseased aorta and replace it by Dacron graft. While this replacement of the entire aorta may seem ideal for patients who have tears in the aorta, the complications involved are too high (M. R. Moon et al. 2001; Marc R. Moon and Miller 1999). Few surgeons also practice alternate repair method which involves replacement of only that region aorta which comprises of primary tear by Dacron graft and leave the remaining aorta with secondary tears intact. This method reduces the complications involved in previous method. So while researchers are certain about the involvement of the arch with ascending acute aortic dissections (Type-A dissections), there is a controversy on the necessity of replacing the aortic arch during replacement of the ascending aorta (Marc R. Moon 2009; Marc R. Moon and Miller 1999; Urbanski et al. 2010; Geirsson 2010). A schematic of a case where an entire aorta was replaced by Dacron graft has been shown in Figure 15(a) and another schematic where only ascending aorta was replaced by Dacron graft in Figure 15(b). A retrospective study conducted by our group found that involvement of the arch with type-A dissection does not justify total arch replacement (Shrestha 2012). As mentioned earlier, in the current study we have used a CT scan of a Type-A dissection where the primary tear is at ascending aorta and dissection extends from ascending aorta to abdominal aorta. There are altogether 5 tears in this dissection model which can be seen in left hand side of Figure 46 and Figure 47. A repair model mimicking the repair surgery was prepared by patching the primary tear from the dissected model as shown in right side in Figure 46, and a CFD analysis was performed on the volumetric representation of these models to investigate the effectiveness of the repair. This model is termed as Repair-I in this work. In order to further investigate the effectiveness of the method a hypothetical case (Repair-II) was also prepared by removing both the primary and secondary tears from the original Diseased model as shown in Figure 47.

In this study, we quantified the amount of mass flux throughout the cardiac cycle that enter the false lumen of Diseased, Repair-I and Repair-II models. We hypothesized that patching the primary tear in Type-A dissection will considerably reduce the flow of blood in false lumen hence

decreasing the possibility of propagation of tear both proximally and distally. We also hypothesized that this will also restore the blood flow in true lumen and hence throughout the body. We also sought to determine the extent to which Repair-II, with all tears patched/repaired, more effectively restored blood flow in the true lumen, to determine the relative benefit to this more time-intensive and complex repair route. Hence, our discussion focuses on a comparison of these three cases in order to help quantify the relative effectiveness of repair surgery represented by Repair-I model.

Figure 50 shows a qualitative comparison of velocity vectors at peak systole, scaled and colored by magnitude at the same cross sections for all the three models. We considered those cross sections for analysis because we were interested in examining the hemodynamics in the vicinity of the proximal and secondary tears. It is apparent that both Repair-I and Repair-II models exhibit a significantly reduced mass flow in the false lumen. While the reduction is comparable in arch region for both repair models, the reduction is more pronounced in the descending aorta for Repair-II. This shows that secondary tear in the descending aorta of Repair-I brings in blood from the true lumen through the secondary tear. At the entrance of the secondary tear in the false lumen, we see a localized region of recirculation. Blood entering the false lumen is seen travelling downward, towards the abdominal aorta. Hence in the next step we quantify the mass flux throughout the cardiac cycle for all the three models. Figure 48 and Figure 49 show the comparison of velocity vectors for Diseased, Repair-I and Repair-II models at peak-systole, mid-systole and early diastole. Analysis of cross sections 1-4 show that in these proximal regions, the flow in the false lumen decreases by 99.99% for both the repair cases. The mean mass flow rate in cross sections (1-3) in Figure 50 are about 0.0016 and $1E-04$ cm^3/s , for Repair-I and Repair-II respectively compared to the mean mass flow rate of 16.5 cm^3/s in the diseased model. The effect of the secondary tear is seen in Repair-I as the mean flow in the cross-section 4 (just proximal to the secondary tear) and 5 (just distal to the secondary tear) are 7 cm^3/s in and 15 cm^3/s in the false lumen as compared to 0.0012 and 0.0035 cm^3/s in Repair-II. While this mass flux in Repair-I is higher than Repair-II, it is still lower than Diseased case by 41 % and 18% in the in the respective

proximal and distal cross sections. An examination of cross section 7 (just proximal to the tertiary tear) shows that the mean flow rate in Diseased, Repair-I and Repair-II cases are 36.6, 26.5 and 1.5 cm³/s respectively. This suggests a decrease in flow by about 27% and 94% in the two respective repair cases.

The quantification of mass flow suggests that the patching of primary tear located at the junction of ascending aorta and arch, significantly decreases the mass flux in the arch and cross sections proximal to arch in descending aorta. Though the primary tear is patched, the presence of secondary tear causes mass flux to enter the false lumen. However the increase in momentum is localized in the region of secondary tear and in cross sections downstream. The effect is negligible in cross sections proximal to the arch. We have hypothesized that the propagation of dissection is proportional to mass flux in the false lumen. If this is true, our findings suggest that patching of primary tear decreases the probability of propagation of dissection in the region proximal to the heart. This further implies a decrease in development of cardiac tamponade, aortic insufficiency and coronary obstruction which are the main reasons of immediate mortality of Type-A dissection patients.

The effectiveness of this repair method hence also depends on the location of secondary tears. If a secondary tear is within the arch region then high velocity and pressure in that region may lead to significant blood flow in the false lumen even when the primary tear is repaired. Here, we must note that current CFD analysis assumes walls of the vessel to be rigid. In real life scenario, compliance of the vessel may result in altered mass flux, as well as altered false lumen geometry after repair.

7.4.4.3 Study 3: The impact of location of primary tear within the aorta

In this section, we investigate the effect of location of primary tear in hemodynamics of aorta with Type-A dissection. Four different models of Type-A aortic dissections were prepared by altering the location of primary tear in Diseased model, while also removing the secondary tear (see Figure 57). These altered geometries are named as follows: Tear-Loc1, Tear-Loc2, Tear-Loc3

and Tear-Loc4 (note that Tear-Loc4 is the same as the previously mentioned case, Repair-I, since patching the initial primary tear results in the secondary tear becoming the new “primary” tear). While the primary tear is relocated in these models and the secondary tear has been removed, note the cluster of tertiary tears in the descending aorta remain intact. Figure 54 and Figure 55 show a qualitative comparison of velocity vectors for all of the above models at peak, middle and end of systole. In these figures we can see jet of fluid across the primary tear which impinges on the opposite wall of the false lumen with a peak velocity of about 2.0 m/s for Tear-Loc1-3. In Tear-Loc4, pressure has already significantly decreased before the first entry tear because of the head vessels and an increase in cross section, hence the velocity jet is not as strong with the peak value of 1 m/s. Figure 56 shows a quantitative comparison of mass flow in cross sections (1-6) in these models. The plot of pressure at the entry tears in Figure 57 shows that the pressure with which fluid enters the false lumen depends on the location of the primary tear. Though the arch typically exhibits lower pressure than ascending aorta, the dissected models Tear-Loc1 and Tear-Loc2 have almost the same pressure due to drastic decrease in cross sectional area of the true lumen. As a result, the peak pressure at the primary tear in the true lumen is about 38 mmHg in both of the two cases. For Tear-Loc3 the primary tear is located just after left subclavian artery and the peak pressure at the tear is about 16.7 mmHg, which is about 56% lower. Similarly, Tear-Loc4 has the primary tear further downstream and peak pressure is only 6 mmHg, which is about 84% lower. Hence, more the proximity of the tear to ascending aorta or arch, higher the pressure and velocity of the jet. It can also be concluded that, proximity to ascending aorta or arch increases the likelihood of propagation of dissection.

7.4.4.4 Study 4: Sensitivity study of Impact of secondary tears

A comparison of Diseased case and Tear-Loc1 case allows us to see the effect of the secondary tear on the accurate representation of the hemodynamics in dissected aorta. As mentioned in Study 3, Tear-Loc1 is created by removing only the secondary tear from the Diseased model (See Figure 47 and Figure 53). During segmentation it is likely for secondary tears to get

missed while modeling. This is generally true for small tears and in case of noisy images (Dillon-Murphy et al. 2015). The current study allows us to understand the difference in hemodynamics if we fail to or chose not to model the secondary tear. Computing of mass flux in different cross sections show that the presence of the secondary tear led to a decrease in mass flux in the true lumen by about 11% which is about $78 \text{ cm}^3/\text{s}$ per cardiac cycle in the cross sections distal to the secondary tear. This would certainly be a concern if we were studying hemodynamics with a focus on abdominal aorta and its branches.

7.4.5 Conclusions and limitations on hemodynamics in aortic dissection

To the best of our knowledge, this is the first study of its kind to examine the hemodynamics in Type-A aortic dissection. In this work we have studied the complications introduced by Type-A dissection by comparing hemodynamics in a patient specific diseased model with a baseline healthy case. We found a rise in peak pressure in the true lumen by 12 mm Hg and in an average 30% of the total flow per cardiac cycle got diverted to the false lumen. We found that simulated repair surgery of Type-A dissection by repairing the primary tear to be effective in considerably lowering the amount of blood flow in the false lumen and hence significantly decreasing the chances of backflow of blood towards the heart. The quantity of the blood flow in the false lumen was found to be dependent on the location of the primary tear and the cross sectional area of the true lumen in the vicinity of the tear. Lastly, the effect of the secondary tear was localized and was significant in the cross sections distal to the tear while having little impact on the dynamics of flow in the false lumen proximal to the said tear.

The current study is however not without limitations. The wall of the dissected aorta is modeled as rigid in this work. While blood vessels always have some compliance and it would require an FSI simulation to incorporate wall motion, an aortic dissection condition can be considered comparatively rigid so no motion boundary condition may be acceptable (Ganten et al. 2009; Chen et al. 2013b). We could not obtain patient specific boundary conditions for this study. So we applied pulsatile wave form from a healthy patient. Considering the acute nature of

dissection, this assumption may be acceptable. Further, the same pulsatile boundary conditions was applied for all the hypothetical models. While the boundary conditions are definitely different in before and after repair cases, we assumed that they are relatively consistent just after the surgery

7.5 Conclusions on examples of applications

In this chapter, we have talked about three examples of the questions that we can now ask with the help of the capabilities developed in this work. We investigated the effect of angle of flow extensions and found that the newly added capability of creating flow extensions perpendicular to inlet/outlet helps us create more realistic flow models. We also investigated the difference in hemodynamics in pre and post aneurysm case where the pre-aneurysm case was created by using the capability of removing selected voxels. We found that the shape factors like bottleneck ratio, aspect ratio were too small in this model and hence there was no jet formation which are characteristic of an aneurysm that does not rupture and our results were in sync with the result of the follow up test. Finally, we studied the effect of Type-A aortic dissection by creating permutations of the original diseased model. We found that Type-A aortic dissection lead to high pressure and AWSS in the ascending aorta and arch and that in average 30% of the flow per cardiac cycle was diverted to false lumen in this case. We found repair surgery of Type-A dissection to be effective in considerably lowering the amount of blood flow in the false lumen and hence significantly decreasing the chances of backflow of blood towards the heart. The quantity of the blood flow in the false lumen was found dependent on the location of the primary tear and the cross section of the true lumen in the vicinity of the tear. Lastly, the effect of the secondary tear was localized and was significant in the cross sections distal to the tear while negligible in cross sections upstream.

CHAPTER 8.SUMMARY

The focus of this PhD work was to build a user interactive toolkit that would enable segmentation and manipulation of complicated biological models for CFD analysis. The biological flows that we targeted in this work involved blood flow in aneurysms, aorta and aortic dissections. Different pipelines for segmentation were created to meet the challenges imposed by different geometries. Preparation of segmented entities for CFD required building manipulation capabilities. These capabilities were added in image domain to avoid the complications of working in mesh domain. At the end, hemodynamic analysis of these geometries were done.

Three specific pipelines were developed for preprocessing the object of interest for the final segmentation. The first preprocessing algorithm was a combination of Anisotropic Diffusion filter, Sigmoid filter, Threshold filter. While this pipeline worked well for the segmentation of coronary arteries, larynx, and bladder, difficulties were encountered when segmenting aorta (healthy and diseased). Proximity of bones and presence of spurious connectors were the main challenges in the aortic segmentation. These challenges were overcome by creating another pipeline based on Region growing and morphological filters. Direct implementation of the second pipeline too proved to be inadequate for delineating aortic dissection as it involved complications including but not limited to inhomogeneous intensities, motion artifacts, and small voxel count. This led to building of another pipeline that utilized region growing, morphological and statistical operators for preprocessing. Irrespective of the preprocessing pipeline, the final operations performed for obtaining a smooth levelset representation of the object of interest involved two step levelset based segmentation method which utilizes fast marching and geodesic active contour methods. Results from all the three pipelines were validated against phantom, and manual segmentations.

In order to apply boundary conditions for CFD, segmented entities need to be manipulated. The ends of the vascular entities require extrusion normal to the cross sectional area. For this purpose, capabilities for center calculation using 3D binary thinning algorithm, and normal/angular

extrusion using the calculated centers were developed in the image domain. Capabilities of erasing voxels by specifying a cross section and using sphere and box widgets were also added which are helpful in isolating the object of interest and preparing planar ends for extrusion and applying the boundary conditions. Since the required manipulations are performed in image domain in this toolkit, manipulation steps are taken right after the preprocessing step and before two step levelset based segmentation.

These capabilities not only allowed us to model a variety of geometries like aneurysm, larynx, aorta and even Type-A aortic dissection, but also enabled us to create several permutations of the patient specific geometries. We performed hemodynamic simulations to understand the effect of the geometry on the region of interest. The first analysis showed that result of a flow simulation is effected by the direction of the flow extensions. Hence, in order to get more realistic flow features it is better to use normal extensions in compare to axial extension. The second analysis was conducted on to understand the difference in hemodynamics in a vessel with and without aneurysm. We learnt that it is unlikely for aneurysms with small bottleneck factor and aspect ratio to rupture. The third study demonstrated the significance of modeling aortic root in flow simulation of aorta as root and valves were responsible in creating higher velocities and shear stresses in the root and ascending aorta which are the main areas where dissections are found.

The fourth study was the study of hemodynamic changes introduced by Type-A dissection. This is the first elaborate study of the effect of Type-A dissection up to the best of our knowledge. We found comparatively high velocity and pressure in the collapsed true lumen and primary tear than a healthy baseline aorta. Investigation of the location of primary tear suggested that tears proximal to ascending aorta and arch are in higher pressure causing more blood to pass to false lumen. Hence repair surgery done by replacing the region of aorta consisting of primary tear may be justified. We further analyzed the impact of such repair by patching the primary tear and keeping all the other tears intact. We not only saw mass flux in the true lumen get restored as a result of the repair, but also higher initial pressure in the ascending aorta and arch which would lead to expansion of the true lumen. There was still blood flowing to the false lumen through

secondary tear but since this tear was located at the middle of the descending aorta and the true lumen was not severely collapsed at this region, quantity of mass flux to false lumen was not significant. This finding led to another question- “Is it alright not model secondary tear?” Hence, a study of sensitivity of modeling a secondary tear was carried out. An analysis of the mass flux in the false lumen showed that the hemodynamics in the region distal to the secondary tear gets compromised if the tear is not modelled. Similarly, we would also miss out on simulating the complicated flow features seen in the entrance of the tear in false lumen. However, the impact of secondary tear was not seen in cross sections more proximal to the arch. The Study of the impact of secondary tear and location of primary tear suggests that the magnitude of effect of both the tears depended on the location and size of the tear. This led us to deduce that, the effectiveness of the repair surgery will depend on the location and size of the secondary tear.

In this work we have demonstrated a wide variety of utility of this toolkit. There are many other capabilities which can be added to make this toolkit more versatile and increase its scope of application. Addition of the capabilities to perform segmentation of valve and leaflets, manual segmentation, and morphing are examples of such capabilities which if added will enable us to analyze more interesting and physiologically accurate flow phenomenon. Similarly, addition of a capability of generating tetrahedral mesh will enable our setup to perform validation studies using the conventional methods. Parallelization of this kit is another important aspect that will enable segmentation of images with finer resolution, larger anatomical model with higher computational efficiency.

In summary, we have successfully developed a complete image-based modeling framework with user-interactive capabilities for segmentation and manipulation of patient-specific geometries for CFD analysis. This framework is first of its kind up to the best of our knowledge. This setup enabled us to look into hemodynamics in patient-specific Type-A aortic dissections, which is also the first study of its kind. Finally, this framework has now equipped us with capabilities to model many hypothetical scenarios by manipulating the patient-specific images in hand.

CHAPTER 9. FIGURES

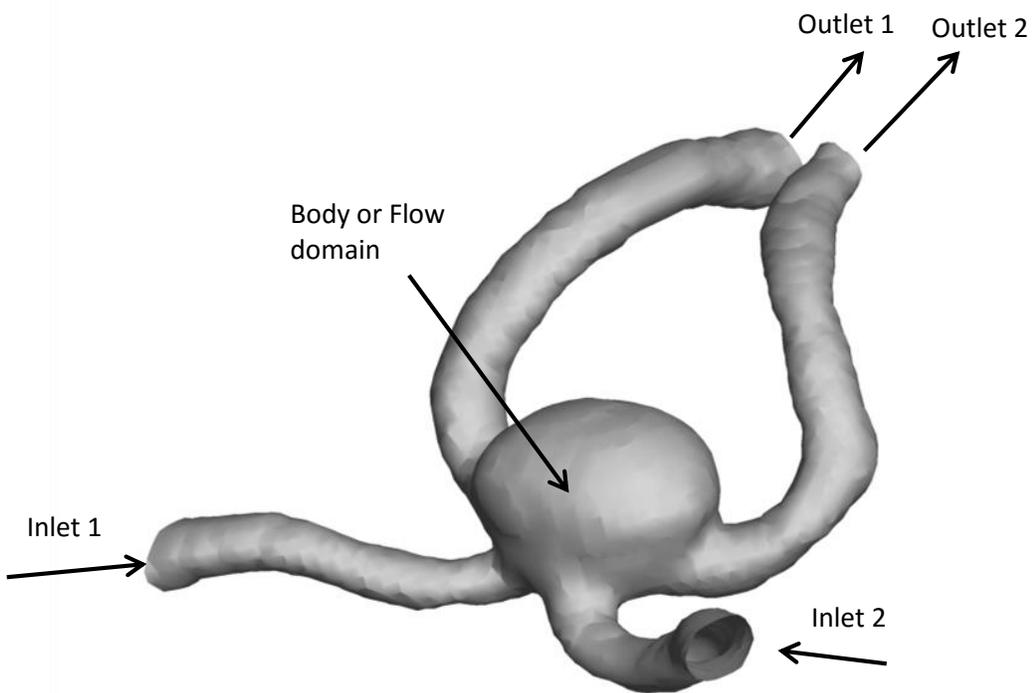


Figure 1 Geometry of Intra Cranial Aneurysm (ICA) showing its two inlets and two outlets and body or flow domain.

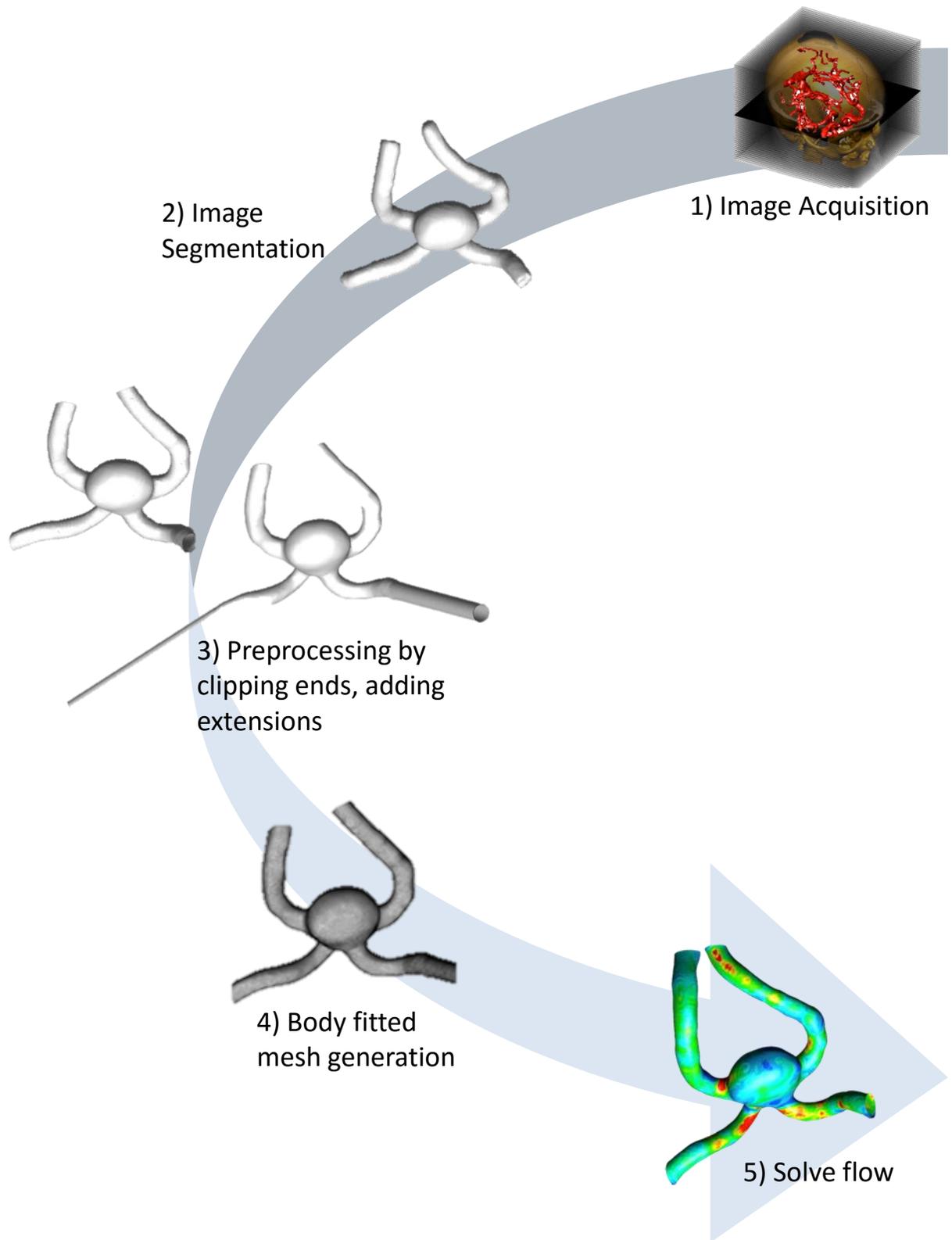


Figure 2 Conventional route of Image based modeling.

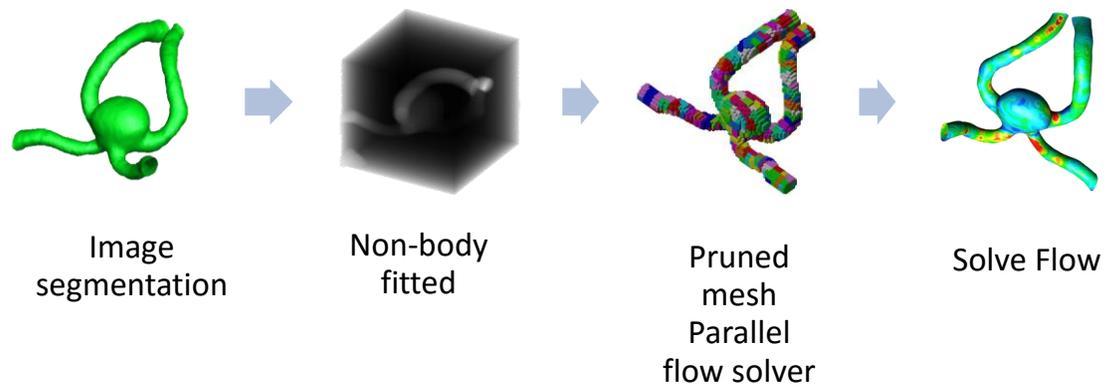


Figure 3 Our approach for image based modeling without body conforming mesh.

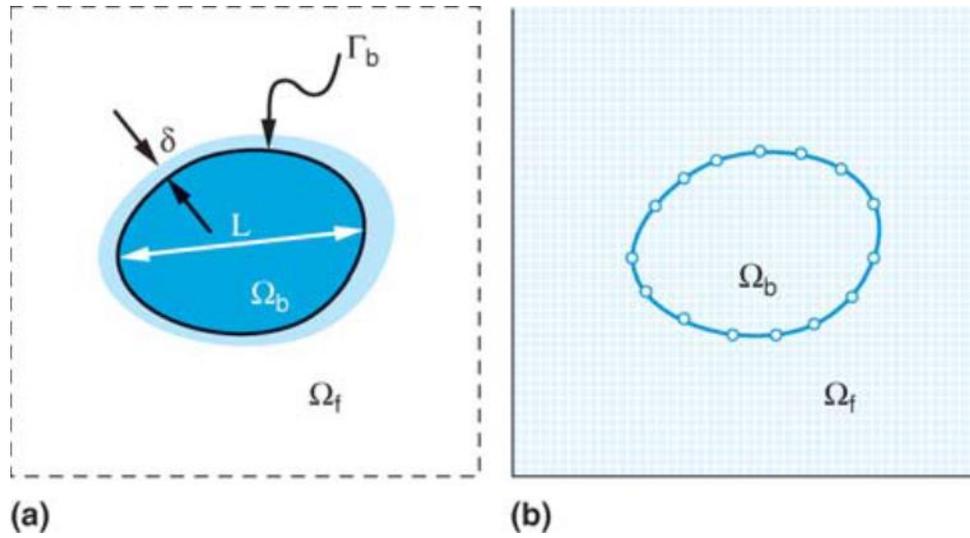


Figure 4 Grid domain of (a) body fitted mesh and (b) non-body fitted mesh

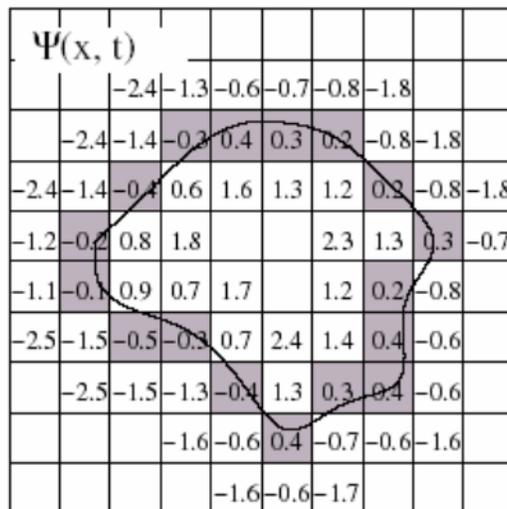


Figure 5 A schematic diagram of levelset field.

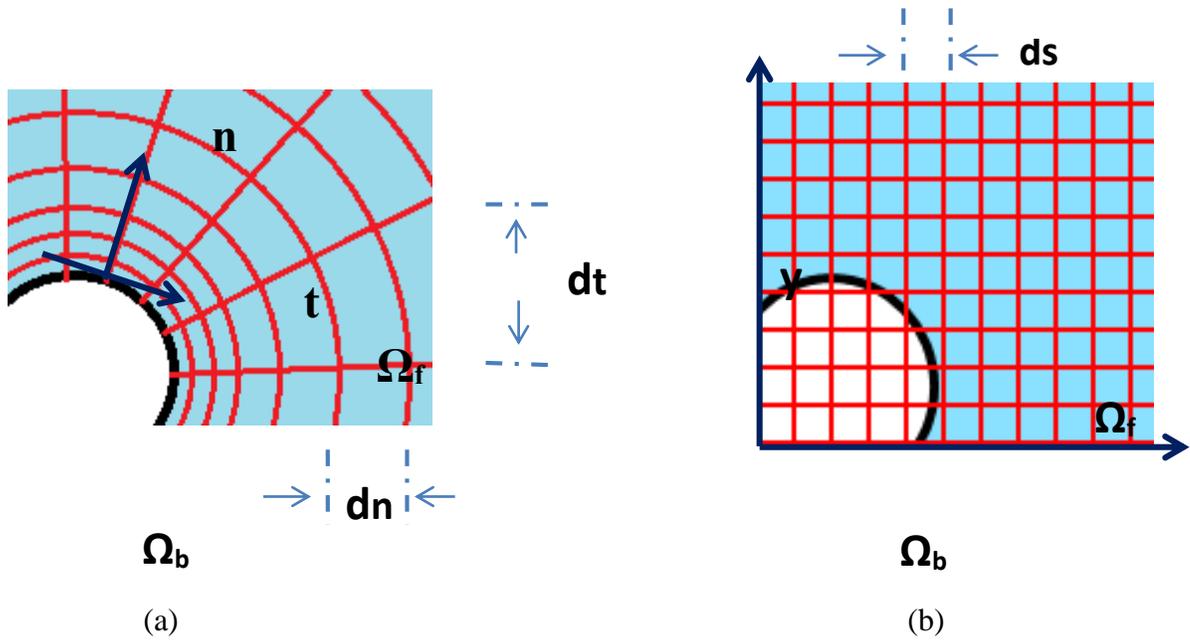


Figure 6 Comparison of grid spacing for the case of flow past a cylinder in two dimensions with (a) body conforming mesh and (b) non-body conforming mesh.

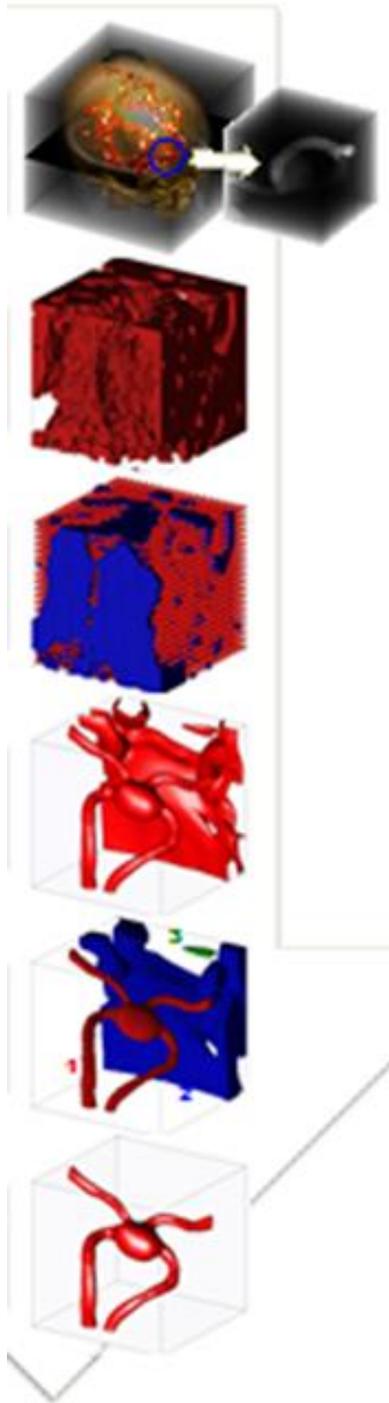


Figure 7 Current route of Image based modeling in our Thermo fluids group based on Active contour without edges.

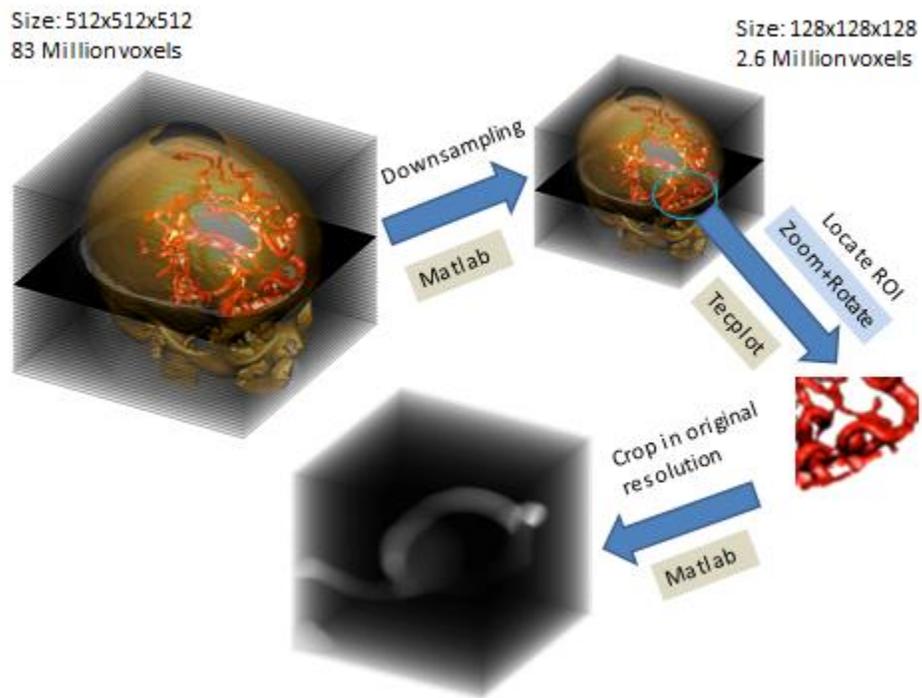


Figure 8 Steps involved in obtaining region of interest from original image in current set up of patient-specific modeling.

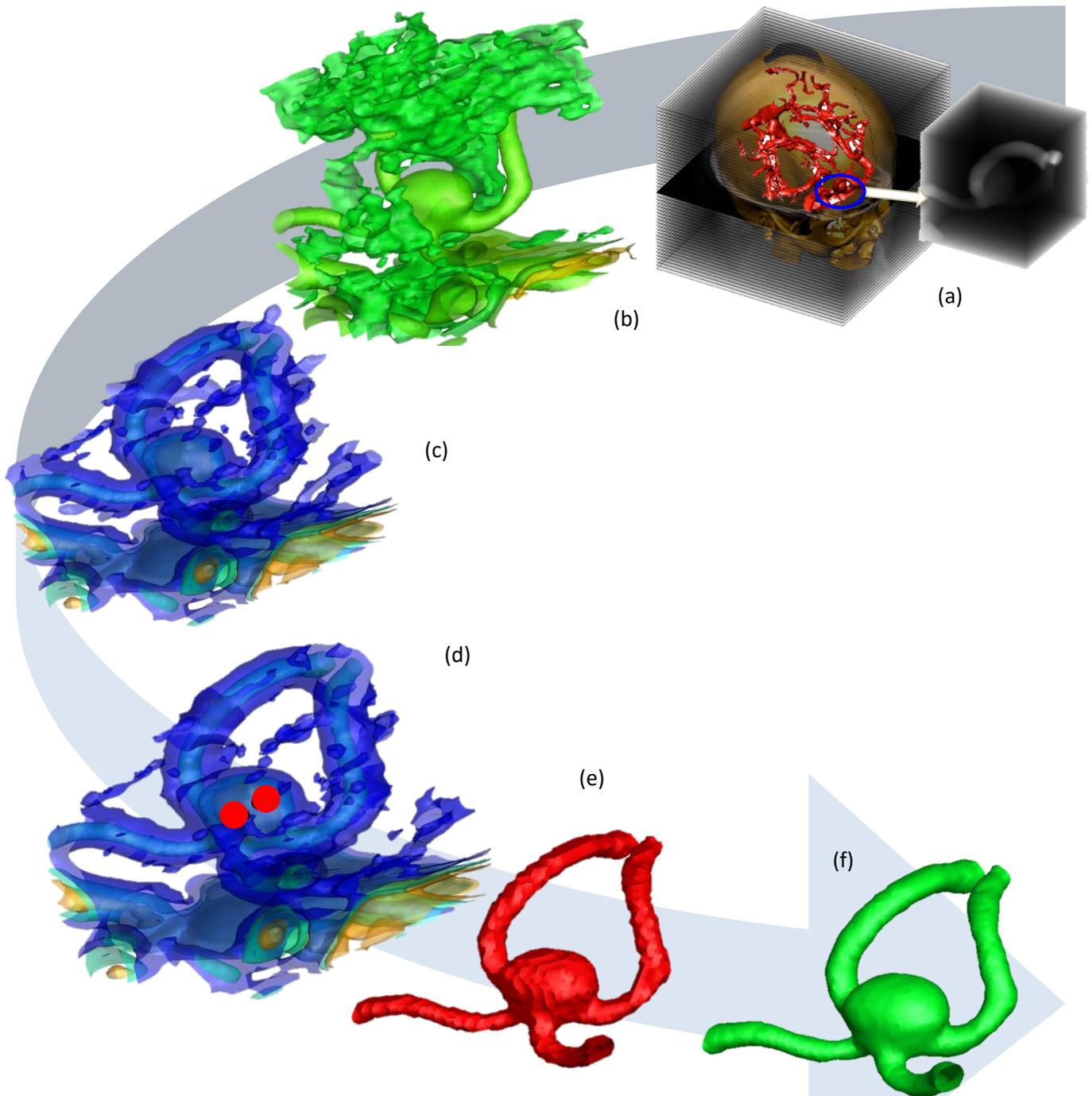


Figure 9 Steps involved in image segmentation (a) image showing Region of Interest (ROI) (b) cropped ROI with pixels scaled from 0.0 to 1.0 colored by intensity (c) result of SigmoidImageFilter highlighting a window of pixel intensities of interest so that the region of interest becomes distinct (d) placing seeds(Red) for FastMarchingImageFilter (e) result of FastMarchingImageFilter used as initial guess for GeodesicActiveContourLevelSetImageFilter (f) zero levelset : result of GeodesicActiveContourLevelSetImageFilter.

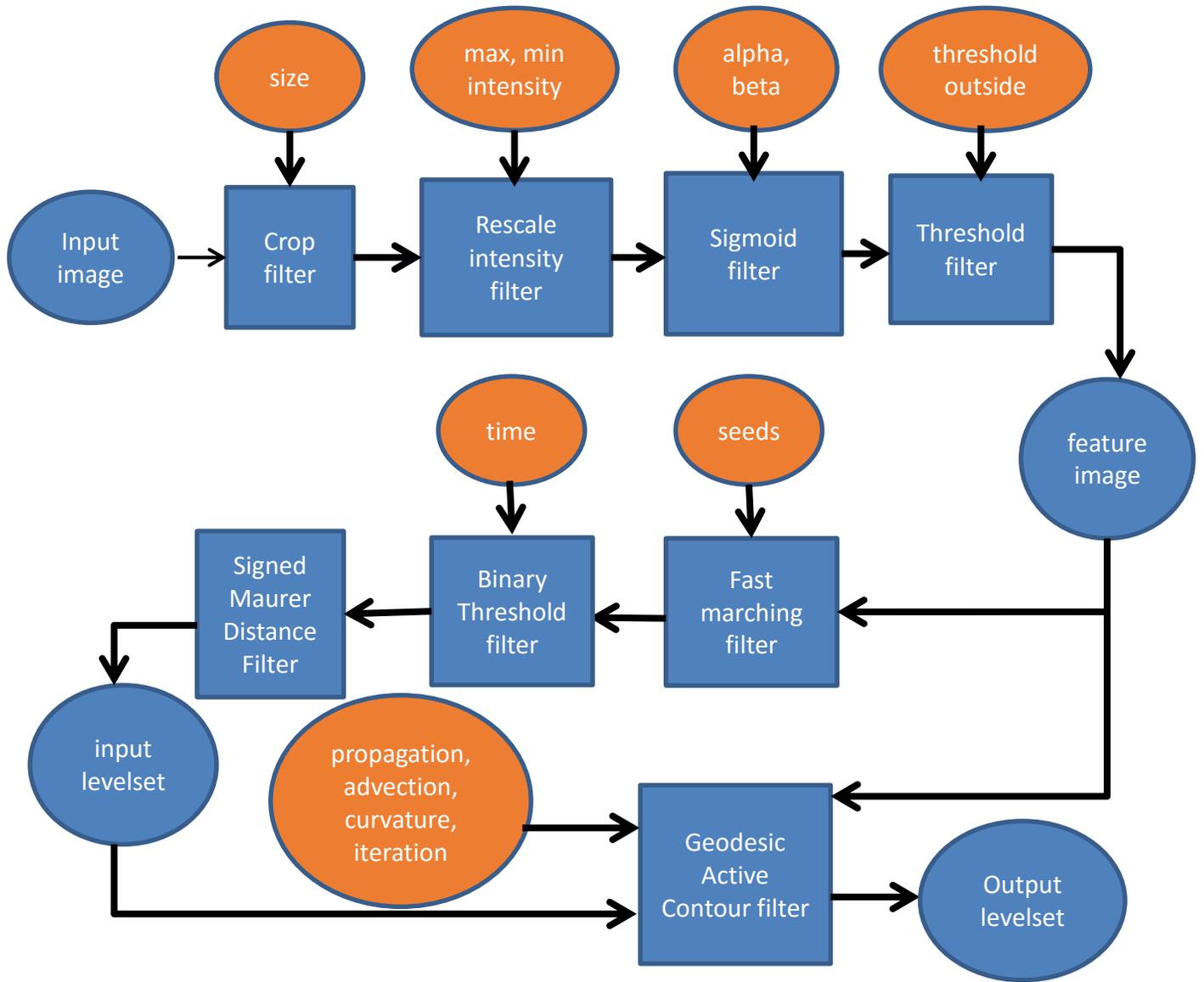


Figure 10 Collaboration diagram of Method I or GeodesicActiveContourLevelsetImageFilter applied for segmentation. The input parameters are placed in orange ellipse, all input and output images are shown in blue ellipse and all filters are shown in blue rectangles.

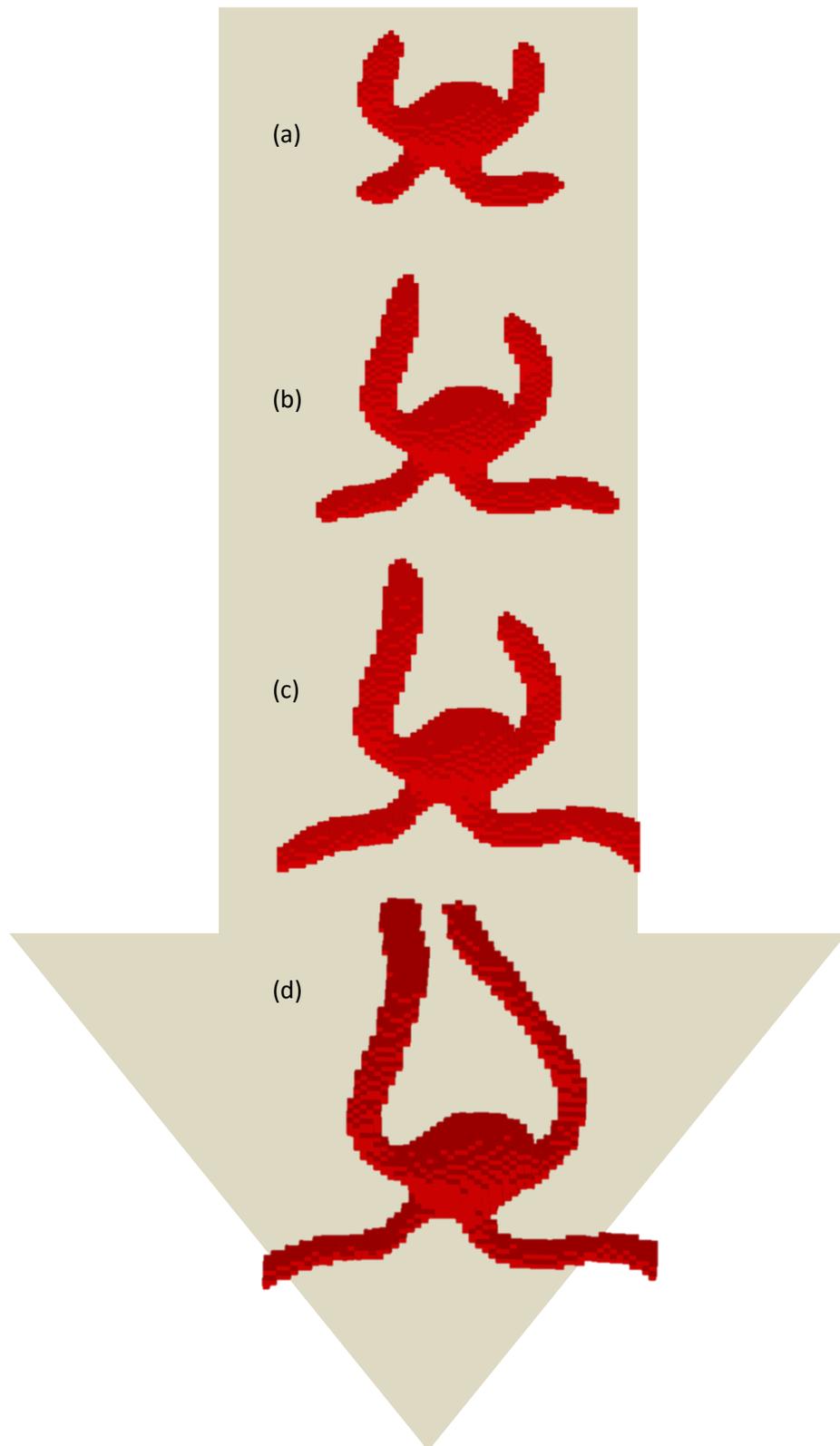


Figure 11 Results of fast marching filter at time steps (a) 60 t, (b) 70 t, (c) 80 t and (d) 90t for Intra Cranial Aneurysm (ICA). User should aim at obtaining result of FastMarchingImageFilter as close to final segmentation result as possible.

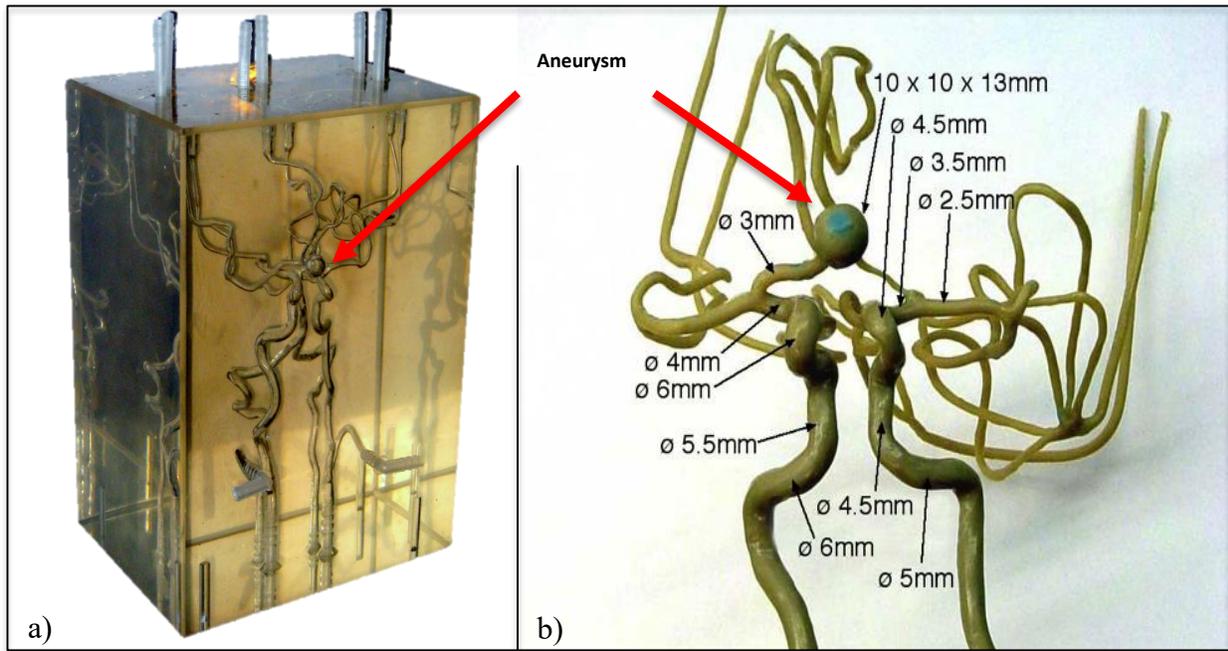
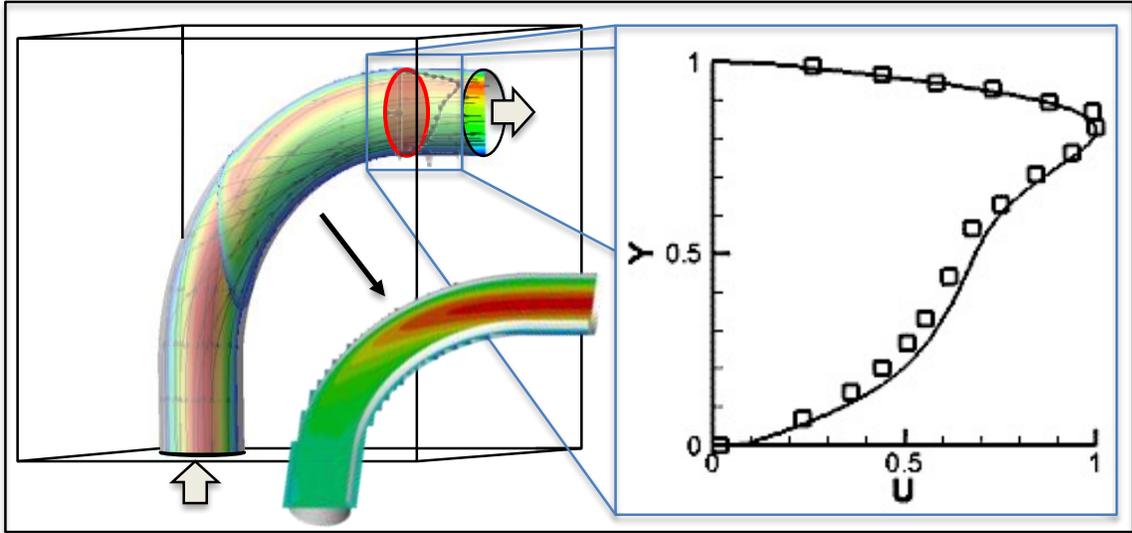


Figure 12 Model of aneurysm obtained from UIHC acquired by normal imaging procedures for aneurysms.



(a)

(b)

Figure 13 Validation of pELAFINT Flow solver with results from paper by (Van De Vosse et al. 1989).

Stanford Type	A		B	
DeBakey Type	I	II	III a	III b
				

Figure 14 Types of aortic dissections: DeBakey Type I (ascending, arch and descending aorta), Type II (ascending aorta only), Type III (descending aorta only). Stanford Type A include Type I and II dissections.

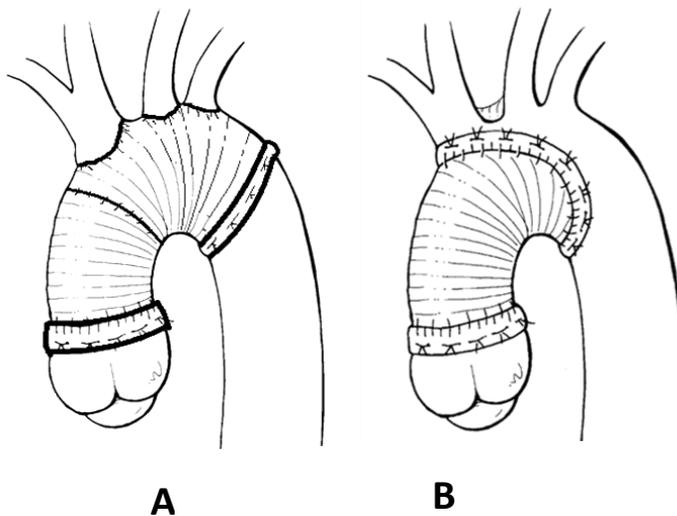


Figure 15 Dacron graft placed after surgery of Type-A dissection in (A) ascending aorta and aortic arch (B) only ascending aorta.

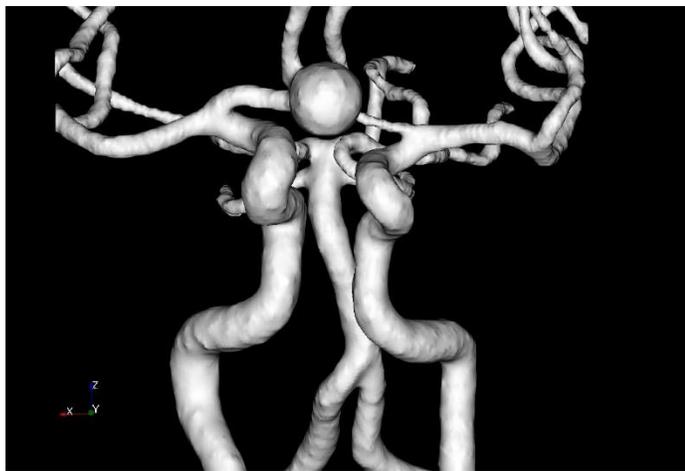
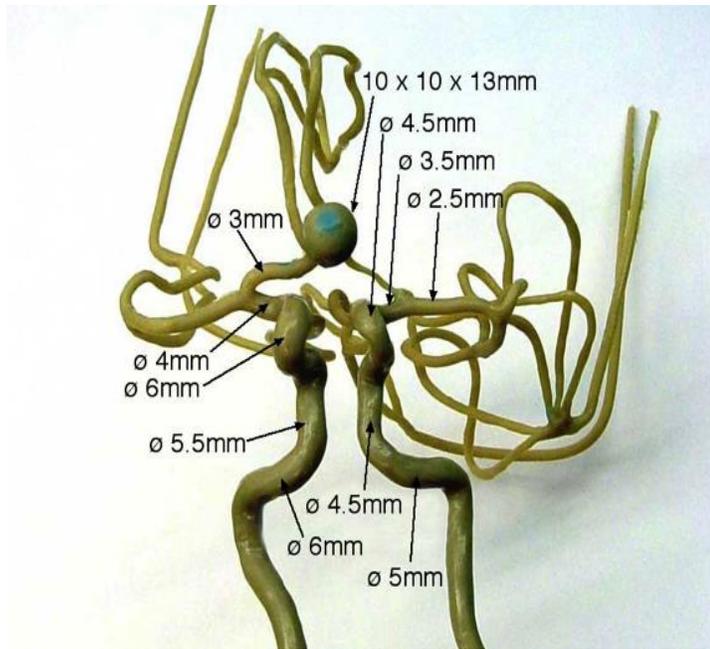


Figure 16 Qualitative comparison of segmented result, (top) original phantom of ICA with dimensions, (bottom) segmented result of the phantom using segmentation Method I.

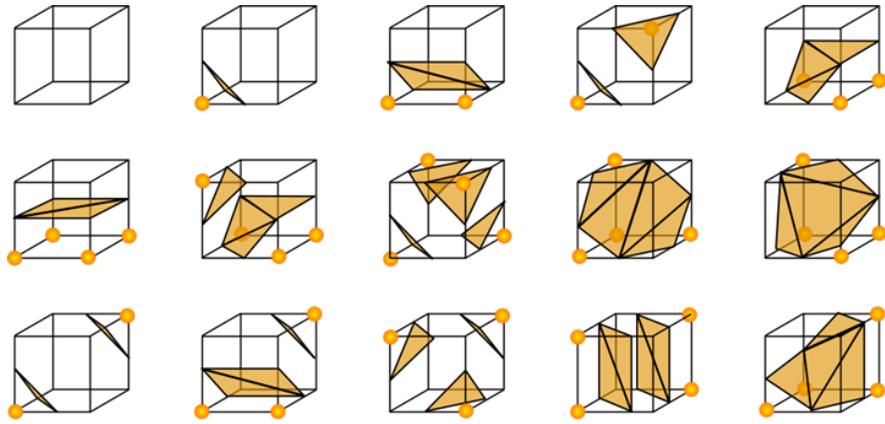


Figure 17 Sixteen standard configurations of the Marching Cubes algorithm. The intersection of an isocontour with the edges of a computational cell leads to arbitrary polygonal shapes. Marching cubes efficiently determines the shape from a hexadecimal lookup table.

Real geometries

Idealized models

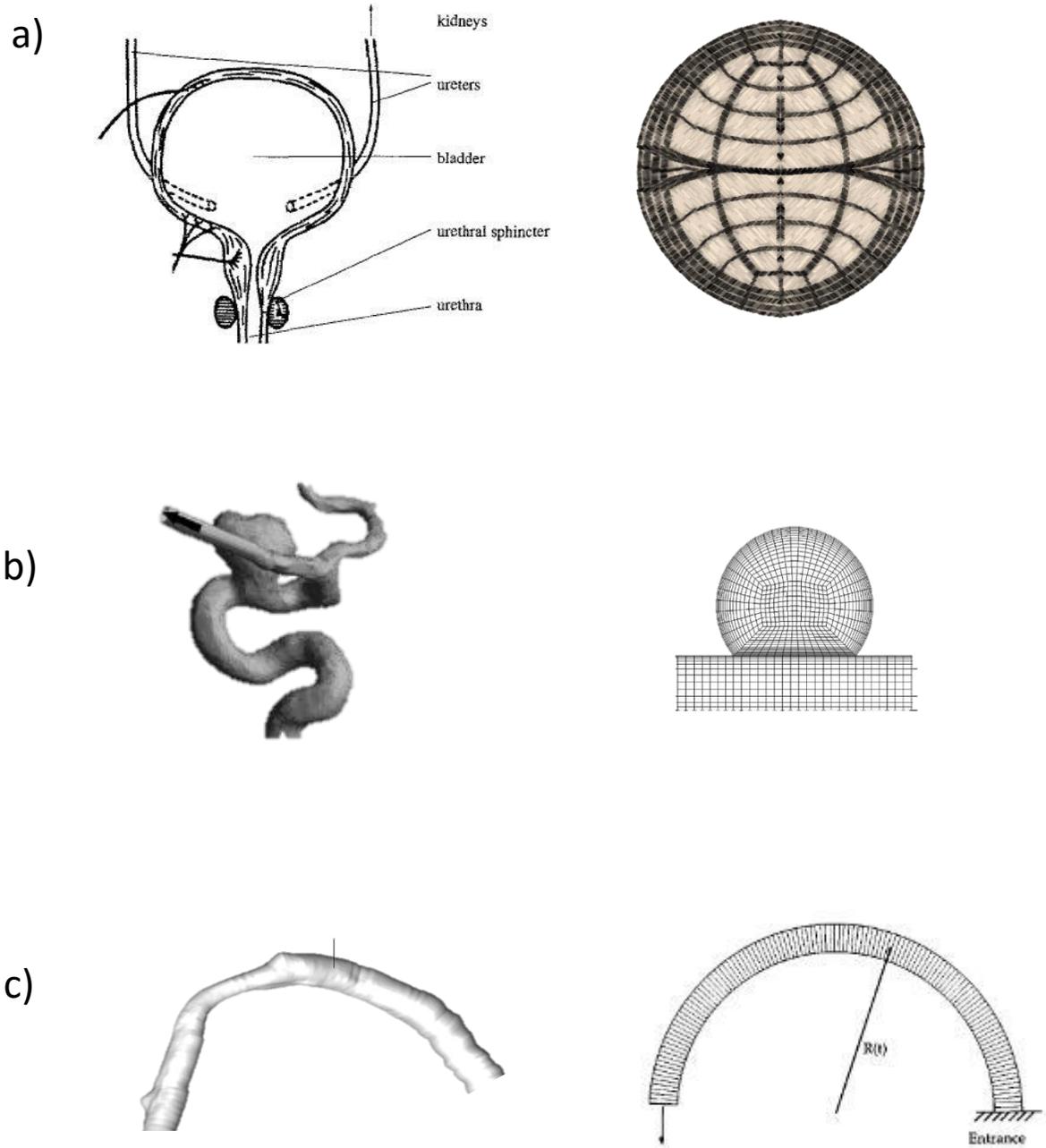


Figure 18 Idealized models used as approximation of real geometries for numerical study of (a) bladder, (b) aneurysm and (c) coronary artery.

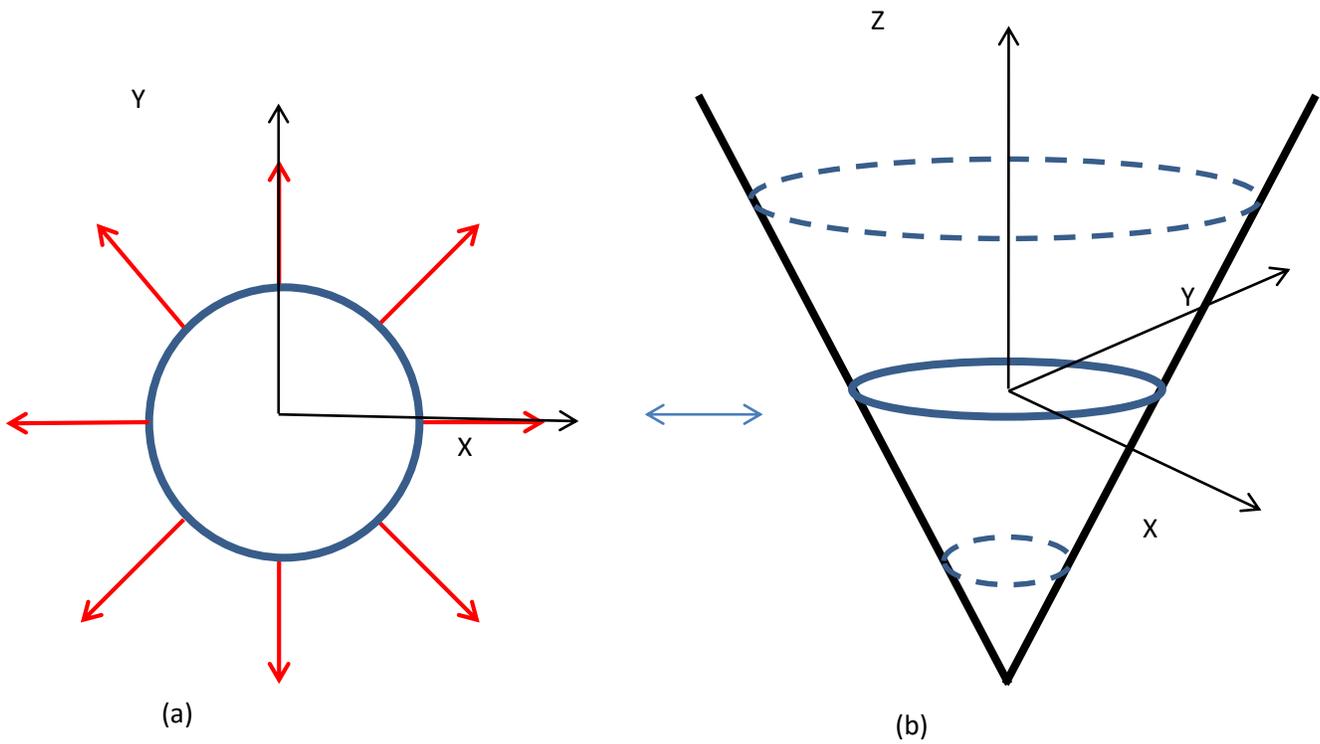


Figure 19 Concept of zero levelset function where we have at (a) a circle representing original front which lies in X-Y plane and is propagating in directions shown by arrow, and at (b) levelset function in the form of cone surface where it shows position of zero levelset as the front propagates. Current position of the front is given by solid blue line and position of the front at different points in time are given by dotted lines.

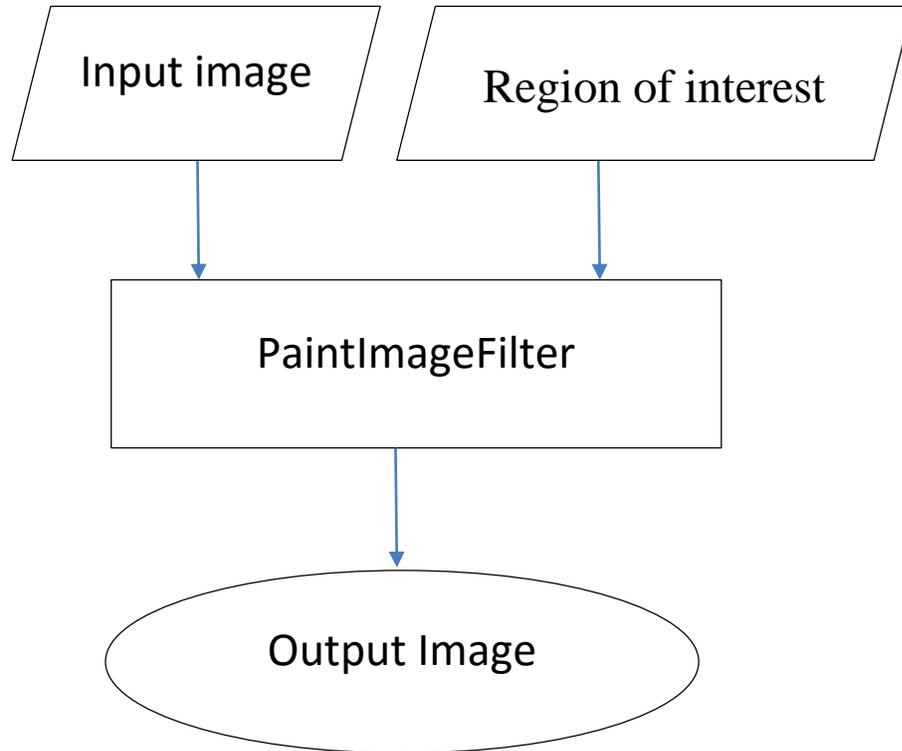


Figure 20 Flow chart of PaintImageGroup.

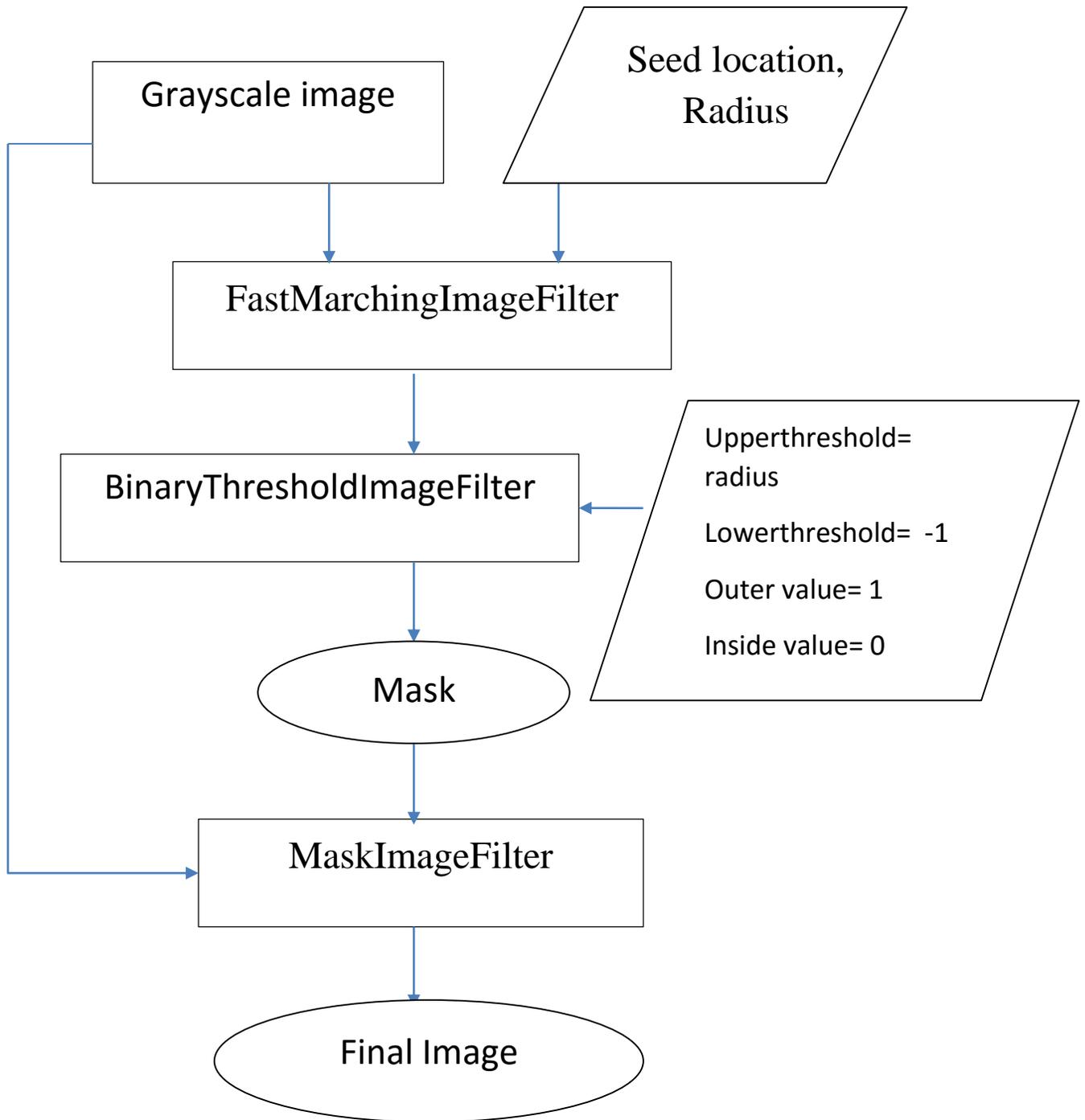


Figure 21 Flowchart of EraseImagePixelGroup.

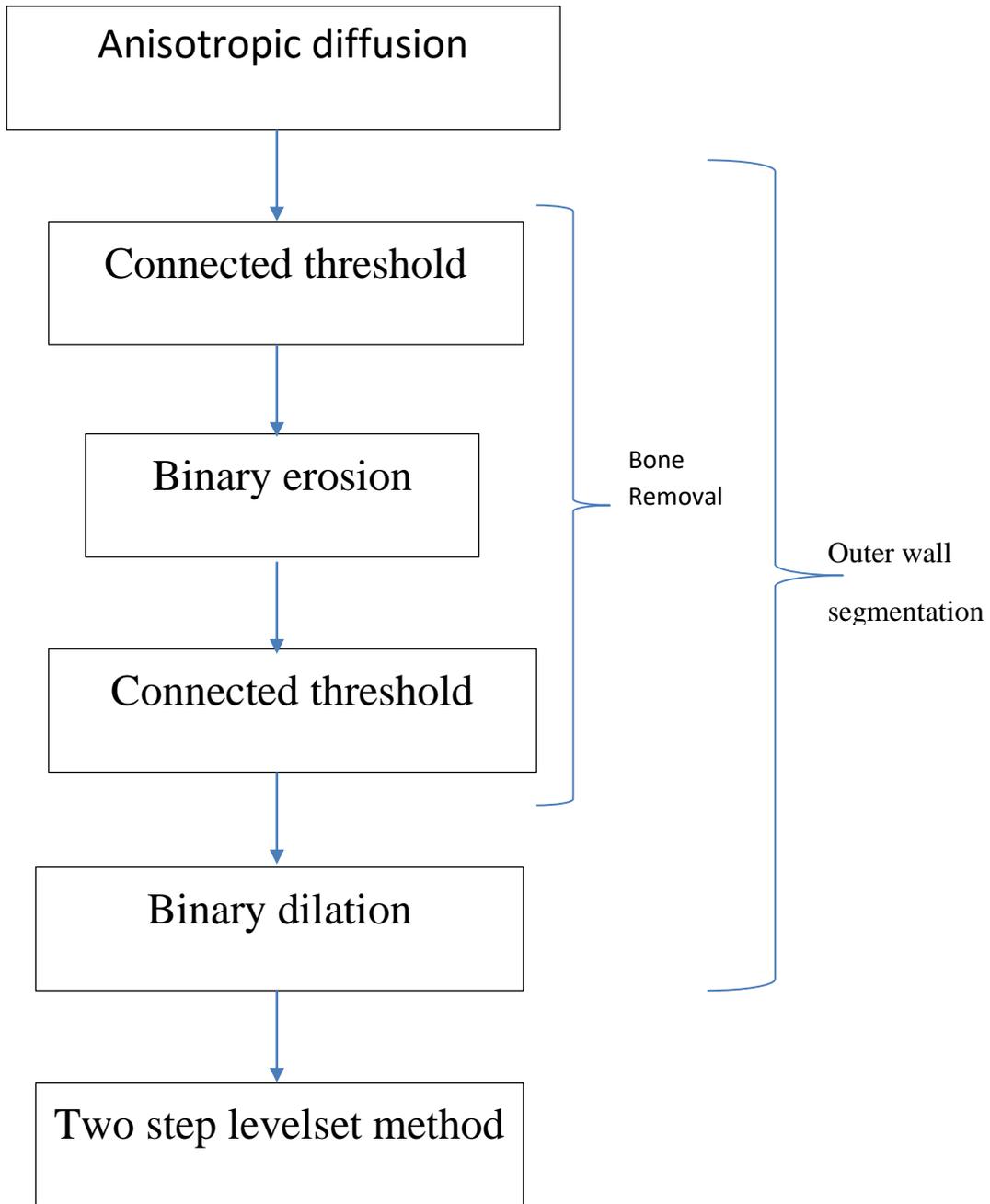


Figure 22 Flowchart of segmentation of aorta or outer wall (Method II).

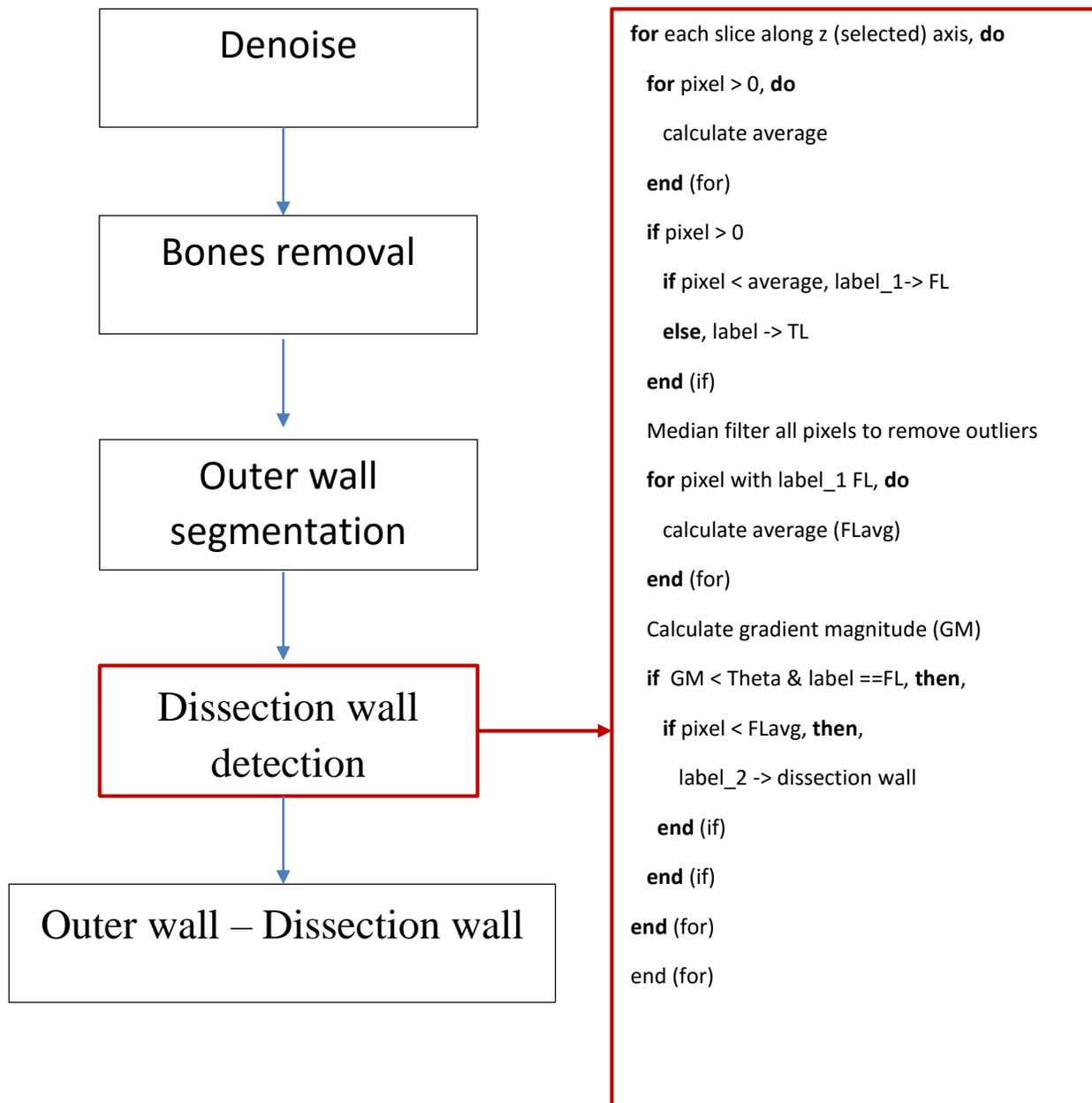


Figure 23 Flowchart of segmentation of dissected aorta (Method III). Algorithm for detection of dissection wall is in inset on right.

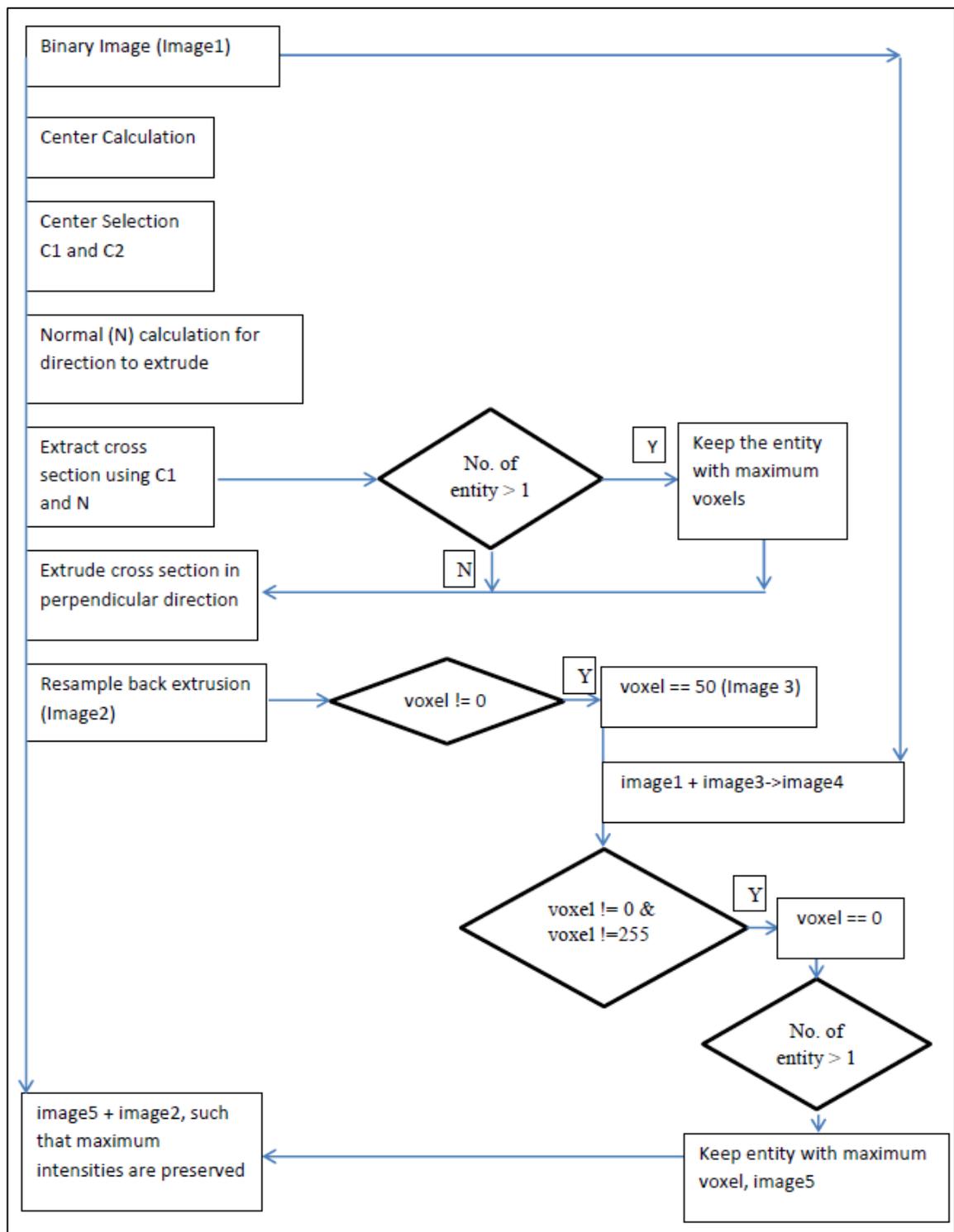
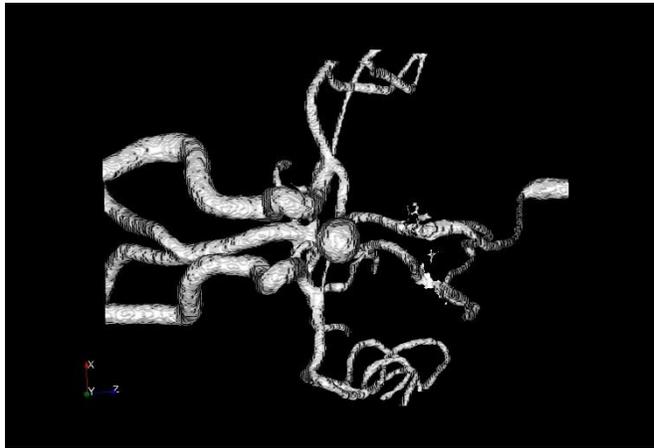
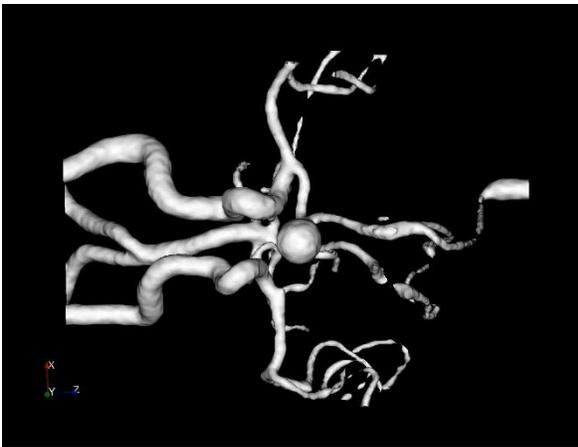


Figure 24 Flowchart of NormalExtrusionGroup.

(a)



(b)



(c)

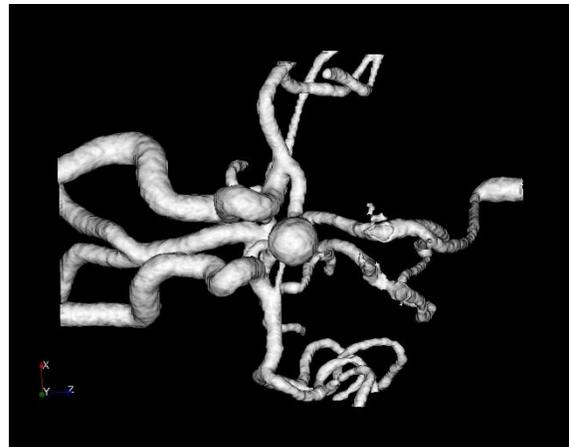


Figure 25 A comparison of results of (a) region growing based segmentation, (b) geodesic active contour segmentation with higher curvature value, and (c) geodesic active contour based segmentation with lower curvature value. The effect of parameters chosen in geodesic active contours for the same input obtained from region growing filter can be seen in (b) and (c).

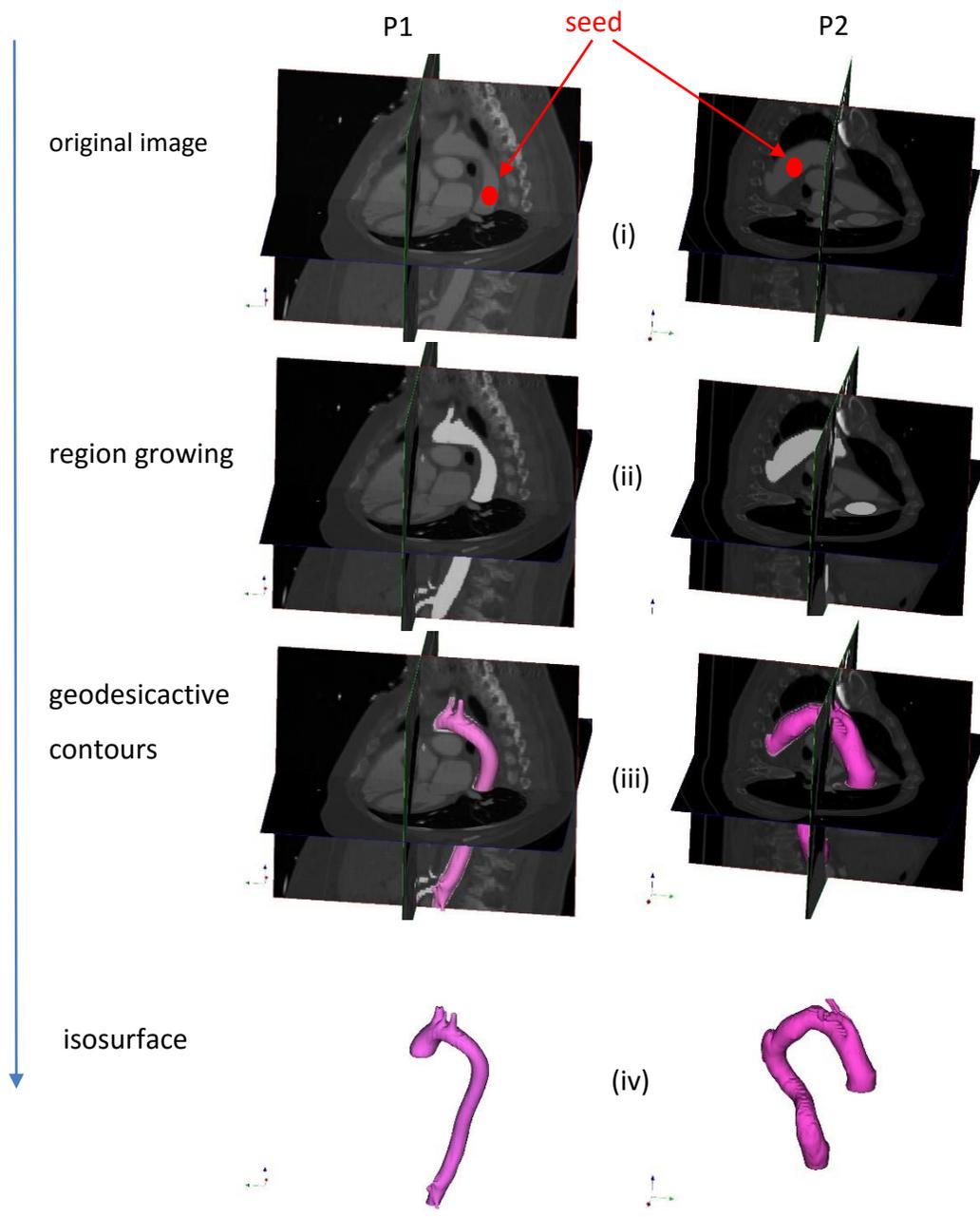


Figure 26 Flowchart for region growing based segmentation of healthy aorta P1 and P2. Original CT scans are shown in level 1, result of preprocessing using region growing method is shown in level 2. Zero levelset obtained from geodesic active contour superimposed on original image is shown in level 3 and final extracted zero levelset is shown in level 4.

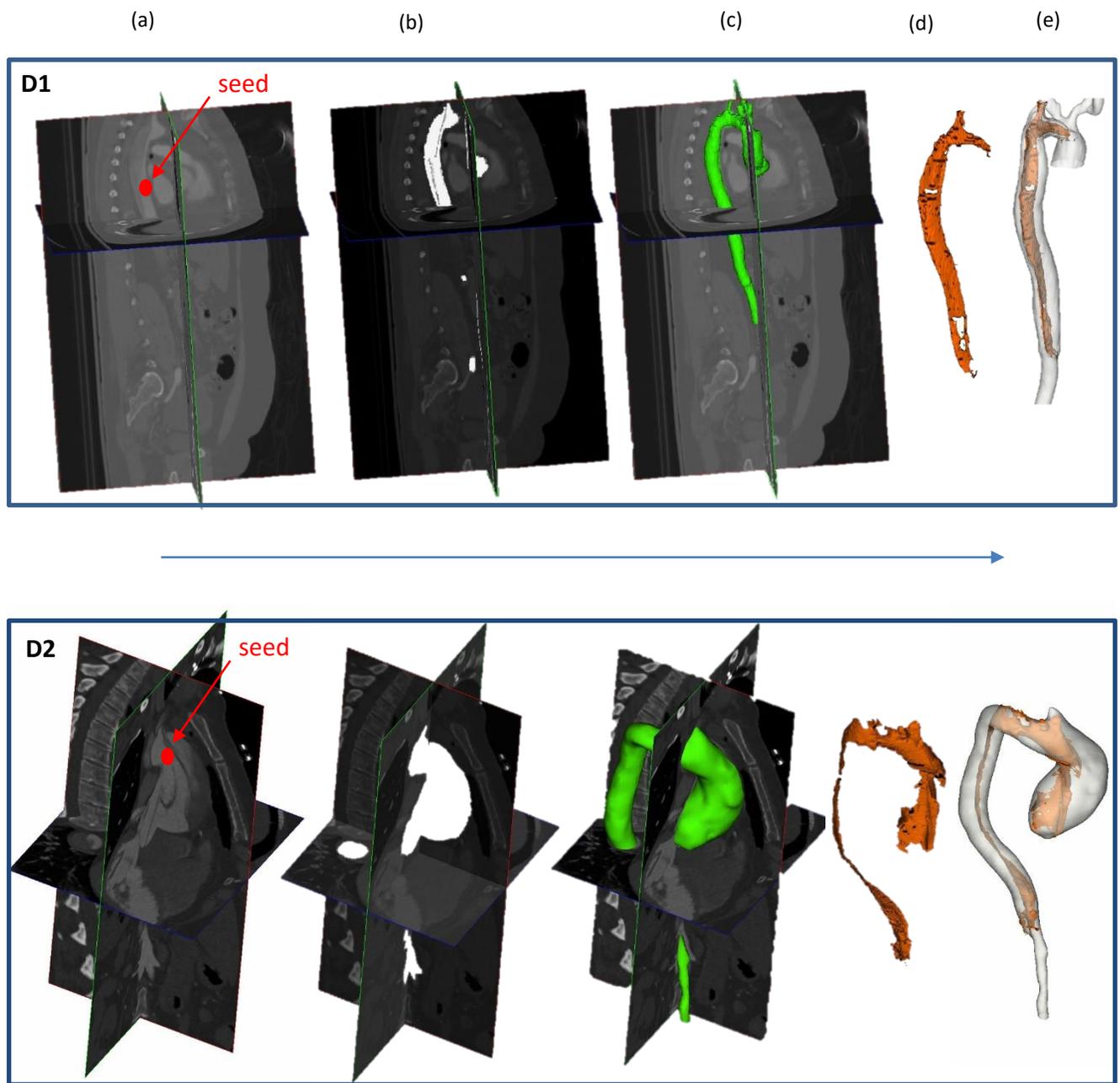


Figure 27 Flowchart showing different stages of segmentation of dissected aortas D1 (top row) and D2 (bottom row). (a) CT scan of a patient with Type-A dissection; (b) image displaying segmented object of interest or outer wall of aorta; (c) zero levelset representation of the outer wall in green; (d) dissection wall in orange and (e) final extracted zero levelset representation of dissected or diseased aorta in white showing dissection wall in orange.

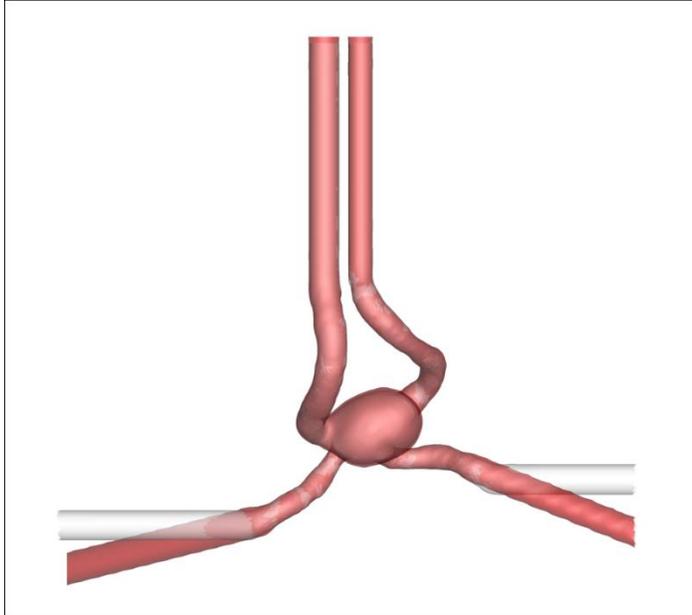


Figure 28 A comparison of two models of ICA created by cutting cross section of inlets and outlets along the axes and extruding (white) Vs cutting the cross sections normal to the geometry and extruding the cross section (red).

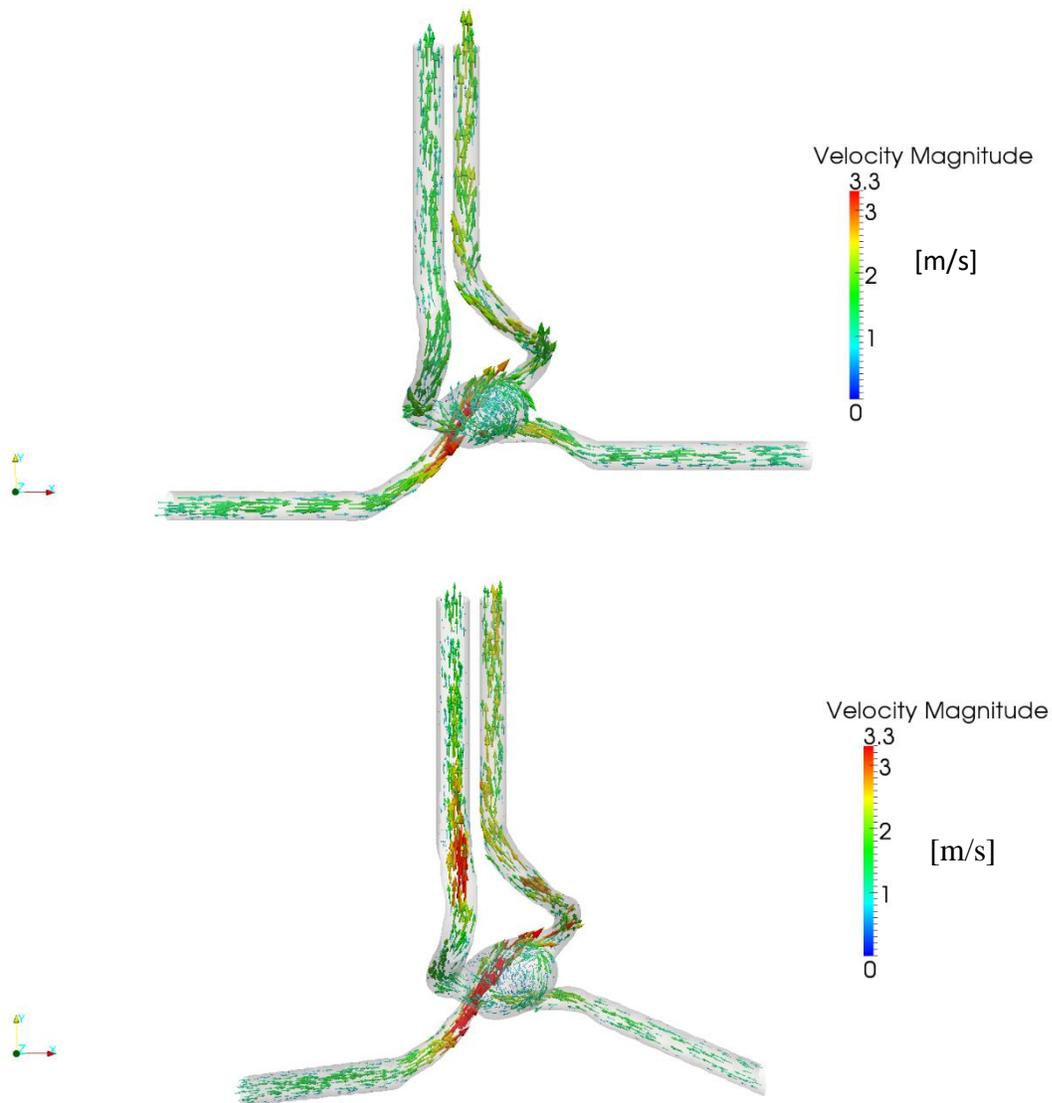


Figure 29 Plots of velocity vectors scaled and colored by velocity magnitude obtained for Model 1 and Model 2, 3d reconstructions of ICA with flow extension normal to the boundaries and parallel to the axes respectively. The maximum velocity is seen at the left entrance to the aneurysm while the rest of the aneurysm has lower velocity and recirculation can be seen.

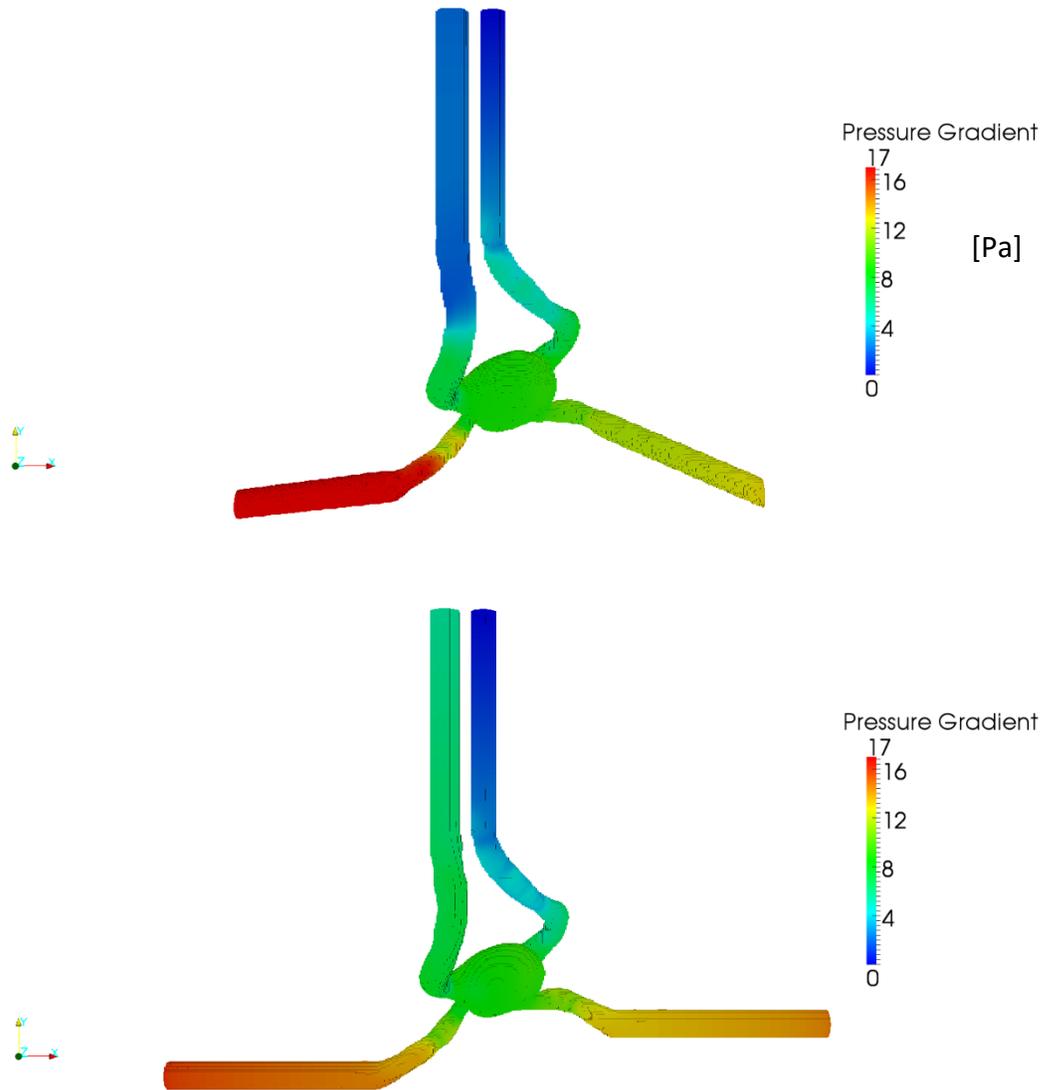


Figure 30 A comparison of pressure in ICA for Model 1 (top) and Model 2 (bottom). The geometries are modeled with flow extension normal to the boundaries and parallel to the axes respectively. Inlet arteries have high pressure of 17 Pascals and outlets have 4 to 0 Pascals.

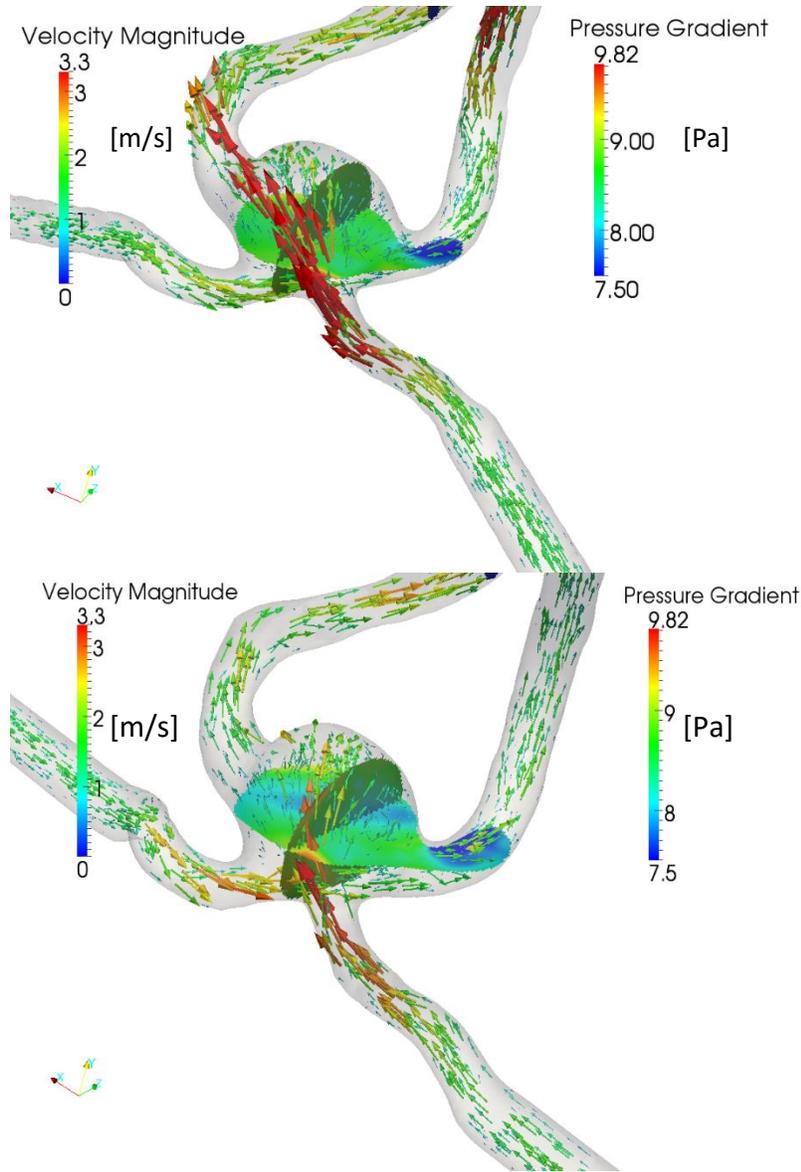


Figure 31 A comparison of pressure distribution on the same cross-sections and velocity vectors colored by velocity magnitude in Model 1 and Model 2 in the study of effect of orientation of extrusions at inlet.

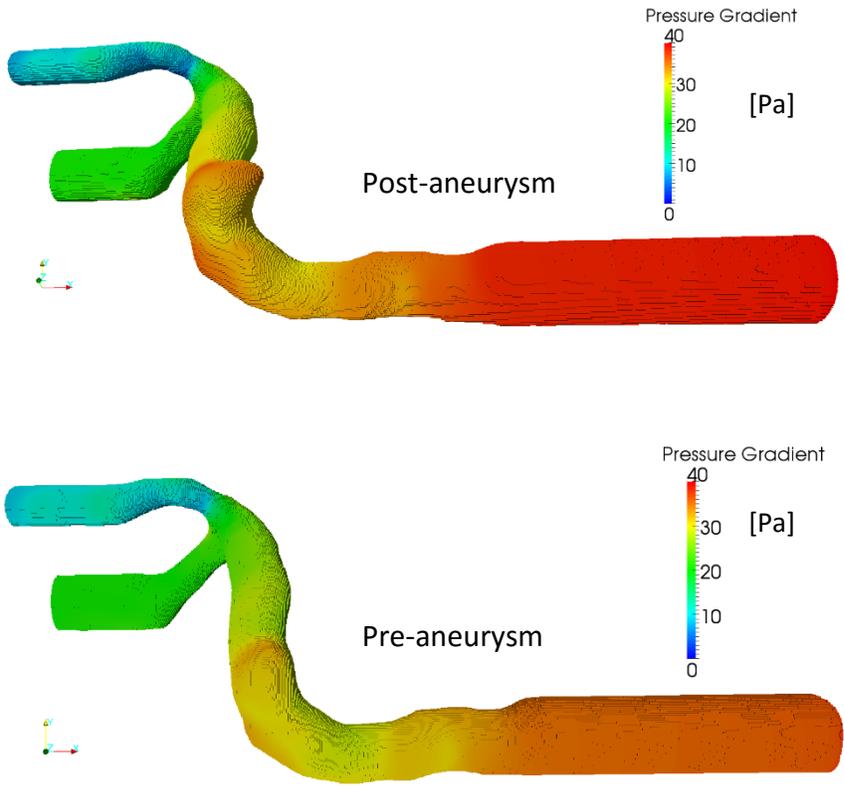


Figure 32 Plot of pressure gradient in Pascals for study of vessel with aneurysm (top) Vs without aneurysm (bottom) under same boundary condition.

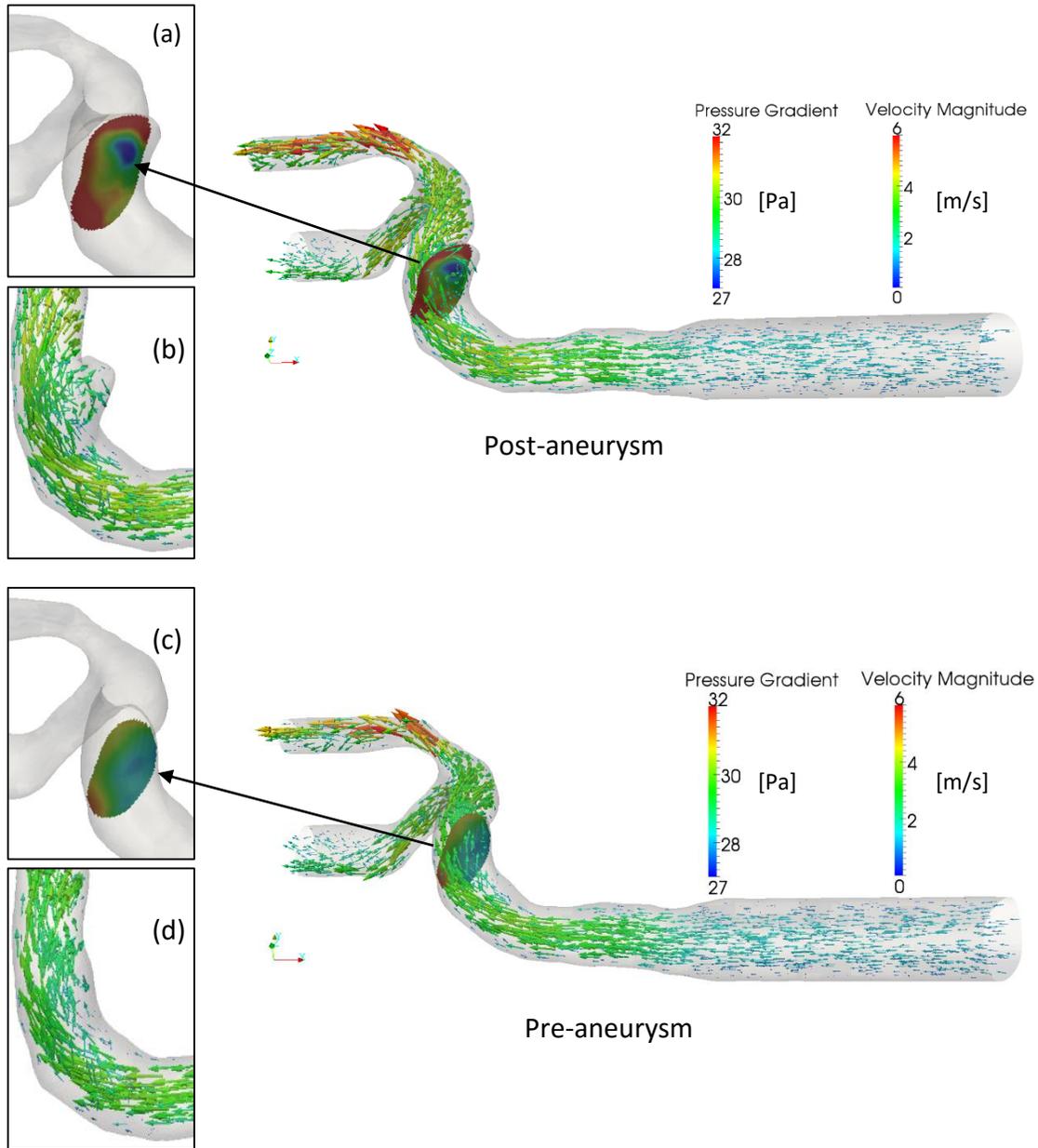


Figure 33 A comparison of velocity vectors passing through a slice, colored by pressure gradient) in the middle of aneurysm and the velocity vectors passing through slice (colored with pressure gradient) at the same position in the model without aneurysm. The insets in the left of each models show the slice and velocity vectors individually.

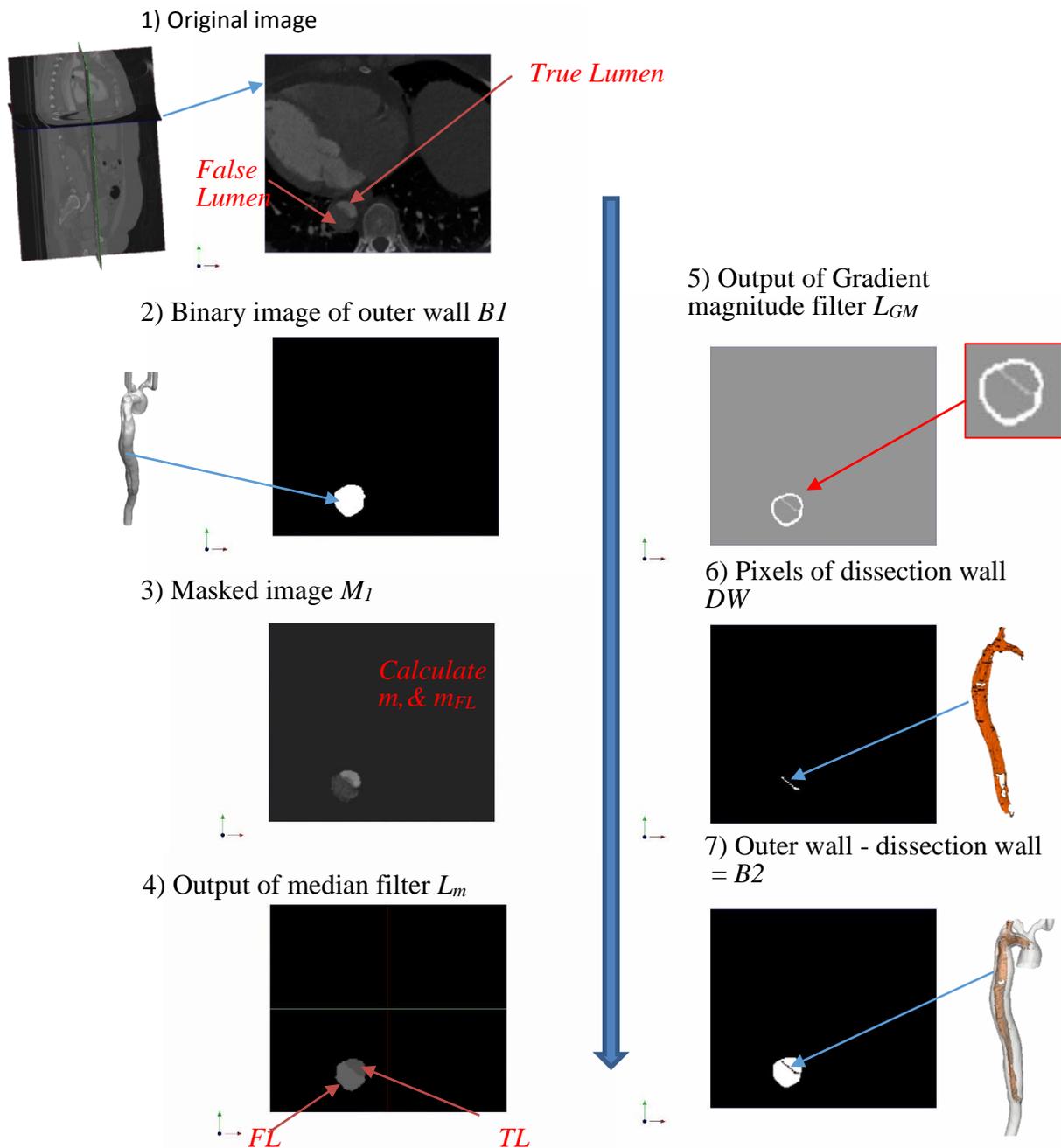


Figure 34 Major steps in segmentation of dissection wall and final dissected aorta (true and false lumen) are shown using a representative slice. 1) Original image, 2) binary mask created from segmentation of outer wall, 3) Original pixels are extracted by applying mask $B1$ on the original image, 4) labeled image obtained by assigning a label FL to pixels below ' m ' and TL to those above ' m ' and applying median filter, 5) output of gradient magnitude filter where the boundary pixels of outer wall have higher intensity and boundary pixels separating true and false lumen have lower intensity, 6) pixels belonging to dissection wall DW are extracted, 7) resulting binary image obtained by removing pixels belonging to DW from $B1$.

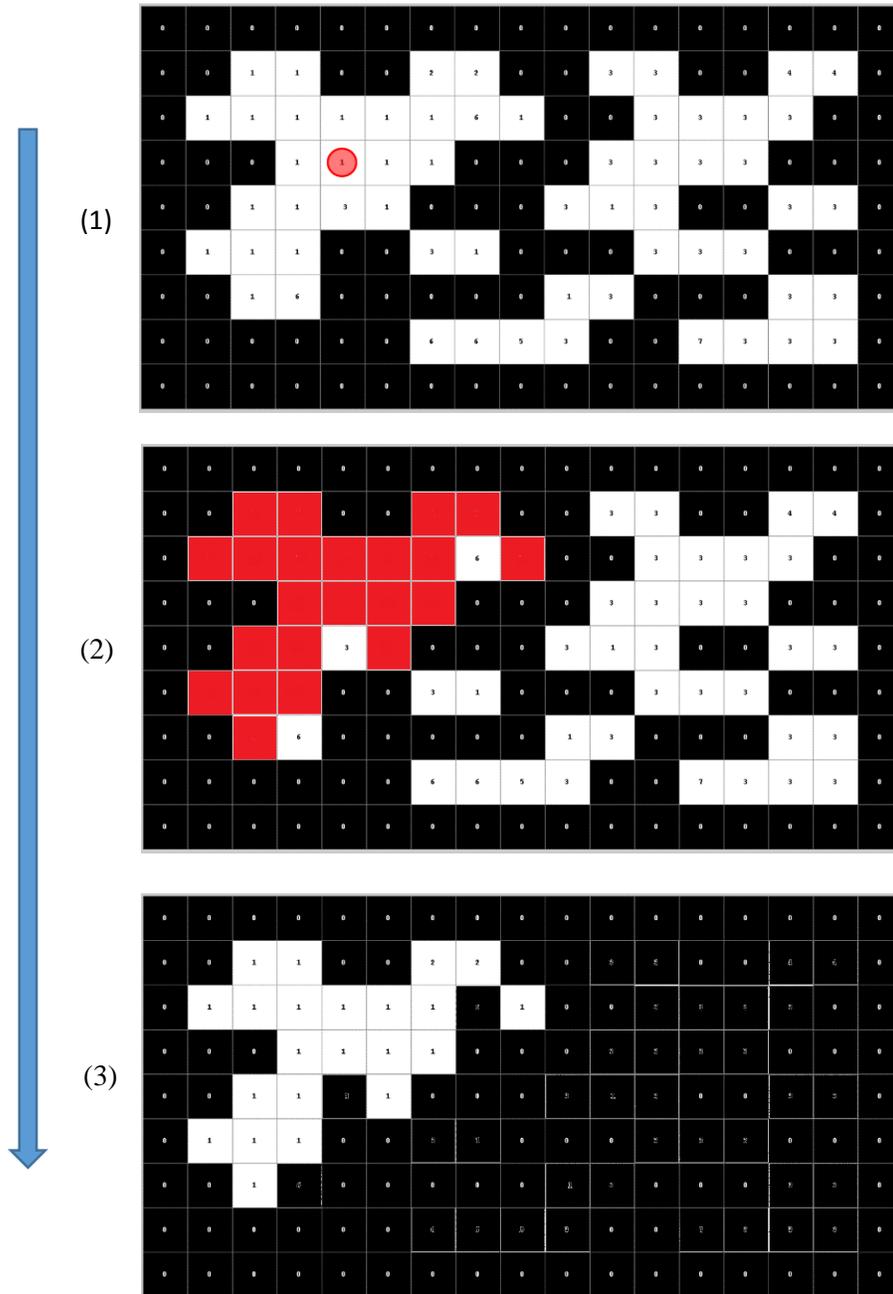


Figure 35 Demonstration of segmentation of region of interest using region growing algorithm in Method II and III. 1) A seed is placed by user in a voxel (here 1) within a ROI that has an intensity which user wants to extract. In this example image the threshold intensities to be extracted are 1 and 2. All the voxels within the threshold connected with the voxel where the seed was placed are labeled (red). 3) The original intensities belonging to the labeled voxels are recovered.

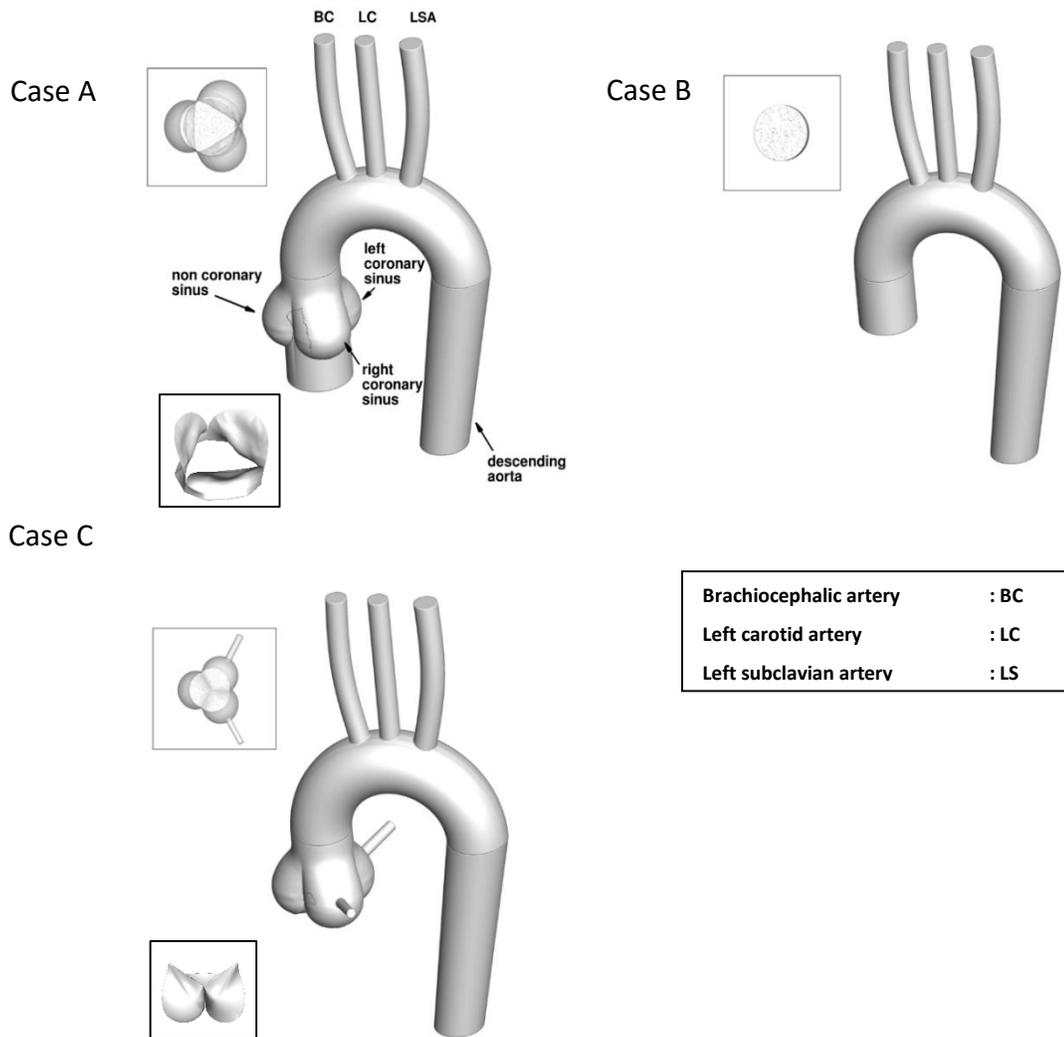


Figure 36 Models created for analyzing the effect of excluding head root and tricuspid valve in hemodynamic analysis. Models of Case A, Case B and Case C, represent the geometry at peak of systole with sinuses and valve (top left), peak of systole without sinuses and valve (top right) and early diastole (bottom). Inserts at the top left of both models of systole show a transverse view of the sinuses and fully open valve forming a triangular inlet to the sinuses for Case A, as opposed to a circular cross section for Case B, without the valve. Insert at the top left of Case C shows the transverse view of the sinuses with coronaries and fully closed aortic valve. Inserts at the bottom left of Case A and C show fully open and completely closed positions of the leaflets of the aortic valve during peak systole and early diastole.

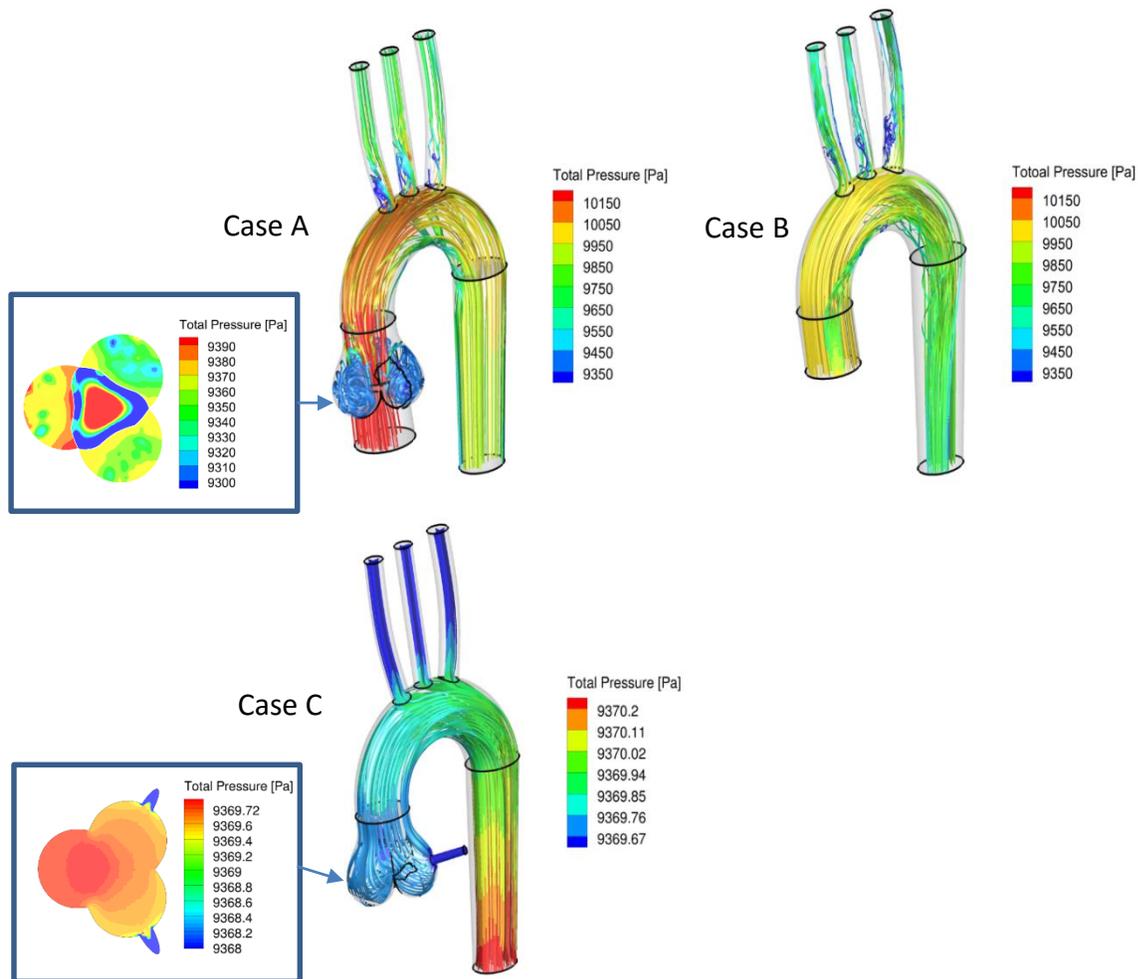


Figure 37 Streamlines colored by pressure for peak systole with sinus and valve (top left), peak systole without sinus and valve (top right), early diastole with sinus and valves (bottom). In Case A, high velocity jet of blood is seen passing through the tricuspid valve aperture directed towards the arch while the blood directed towards the cusps of the sinus is clearly seen decelerated and recirculating. This phenomenon is characteristic of systole and is seen missing in Case B which stresses the importance of including valve and sinus to capture more realistic flow. In Case C, blood flowing with relatively higher velocity and pressure in the arch is seen to gradually decelerate recirculate as it approaches the sinus with valves completely closed. In the insets to the left of Cases A and C, a representative cross section of sinus of Valsalva showing pressure distribution in: peak systole with sinus and valve (left) and early diastole with sinus and valve (right) are shown. In both the cases pressure is higher in non-coronary sinus. This clearly shows that out of the three coronaries pressure is highest in the non-coronary sinus throughout the cardiac cycle.

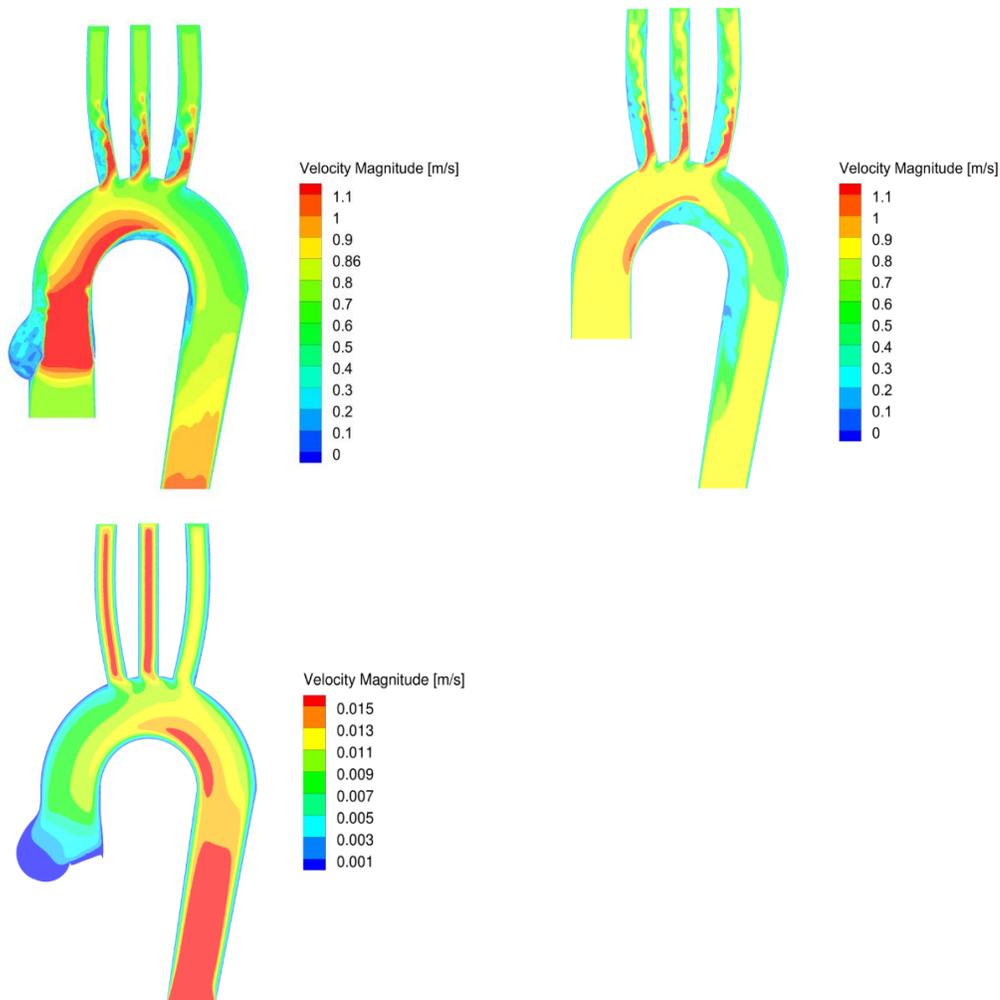


Figure 38 Plots of velocity distribution in the coronal plane during peak systole with sinus and valve (top left), peak systole without sinus and valve (top right), and early diastole with sinus and valve (bottom). The effect of incorporation of sinus and valve aperture is clear in figure of Case A as the high velocity jet adhered to the inner wall is seen directed towards the arch while the blood directed towards the cusps of the sinus is seen decelerated and recirculating. This phenomenon is characteristic of systole and is seen missing in Case B which stresses the importance of including valve and sinus to capture realistic flow.

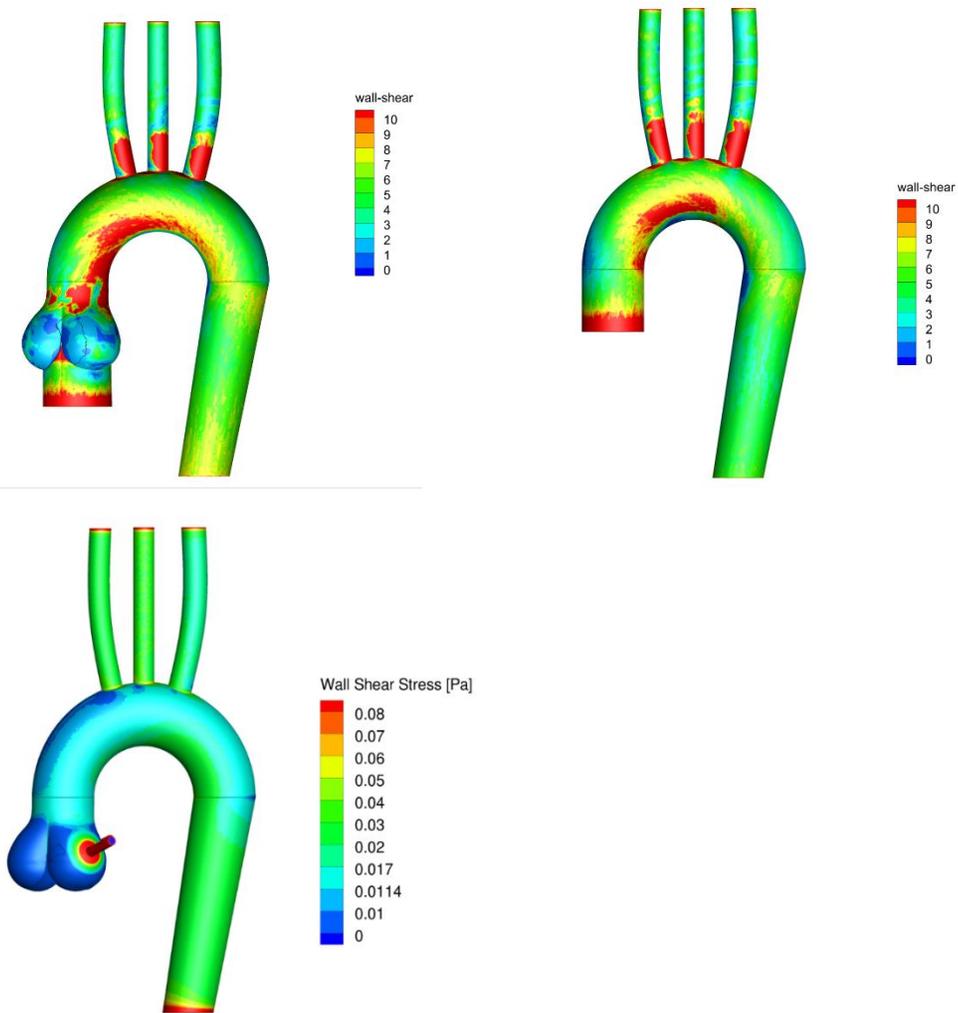


Figure 39 Wall shear stress distribution during peak systole with sinus and valve (top left), peak systole without sinus and valve (top right), and early diastole (bottom). WSS is clearly higher in the inner walls in peak systole as seen in both Case A and B, but their magnitudes and the region of high WSS is more distributed for Case A due to the high velocity jet. Relatively low WSS is seen during early diastole, Case C, due to comparatively lower velocity.

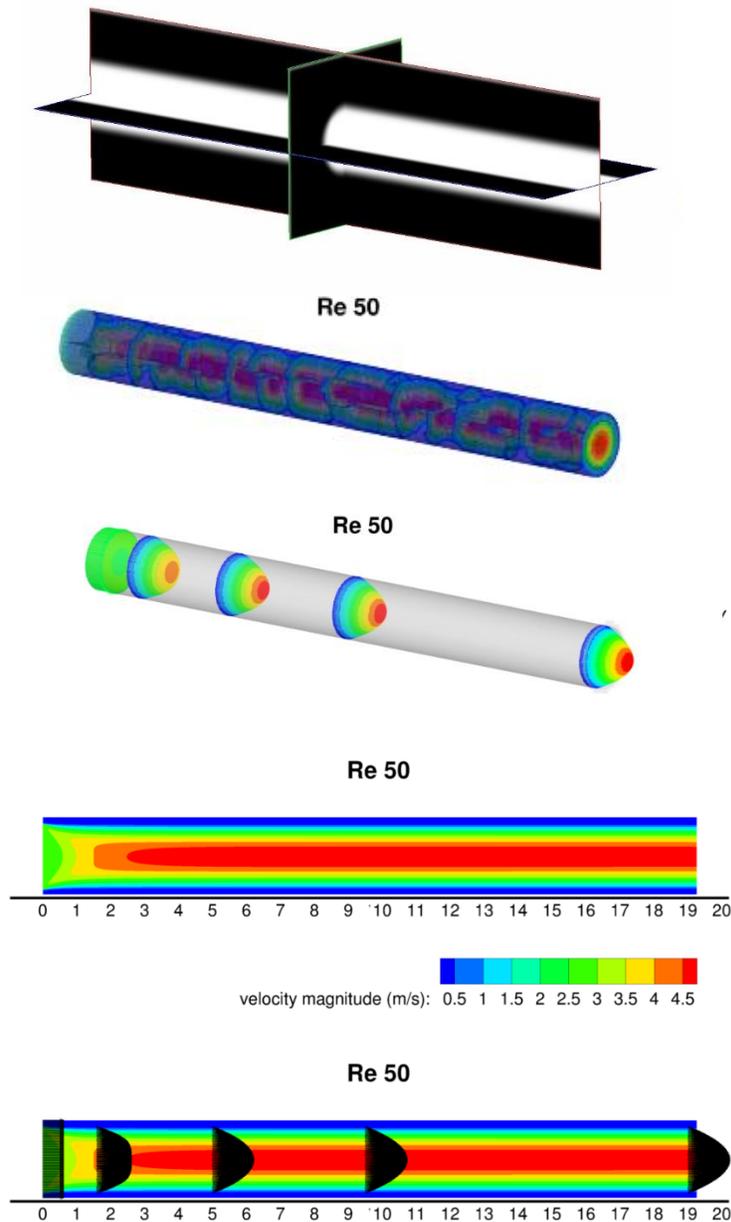


Figure 40 Validation of Image to flow set up by comparing entrance length obtained via simulation against empirical results at Re 50. Binary image of the cylinder (top), rank distribution of flow domain (second), velocity vectors in 3D colored and scaled by velocity magnitude at different locations along the cylindrical domain (third), 2d view of ZY plane colored by velocity magnitude (fourth), velocity vectors at different locations in ZY plane showing fully developed flow at 5 m distance downstream (bottom) which agrees with the empirical result for pipe flow when diameter is 2m and Re is 50.

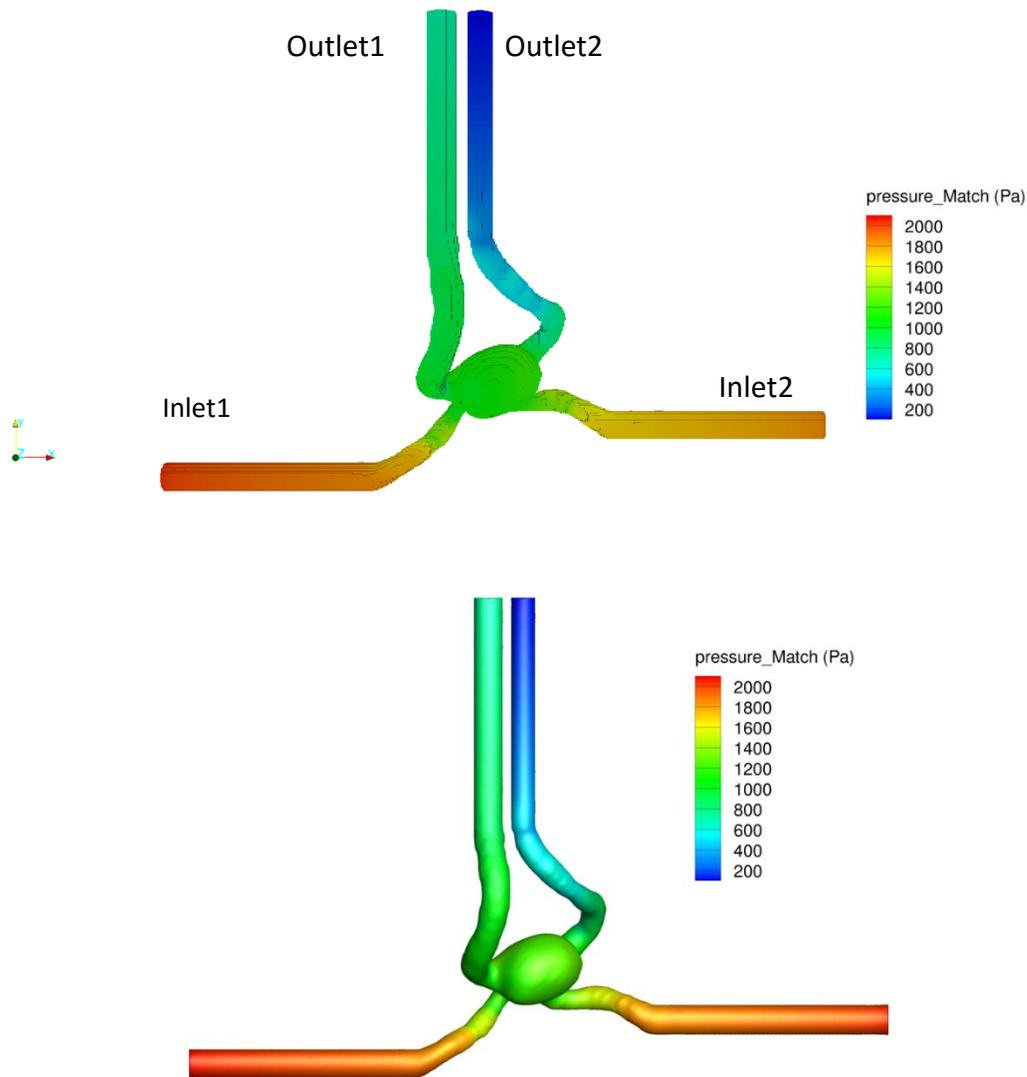


Figure 41 Validation of flow results obtained from image-to-flow set up against flow simulation obtained from FLUENT v14.5.0 (ANSYS Inc., Canonsburg, PA) for intra cranial aneurysm (ICA) for the same boundary conditions of 0.36m/s at left inlet, 0.37m/s at right and 50:50 mass split at the two outlets. Pressure distribution obtained of surface of ICA from in-house flow solver (top) and FLUENT v14.5.0 (ANSYS Inc., Canonsburg, PA) (bottom) are in agreement.

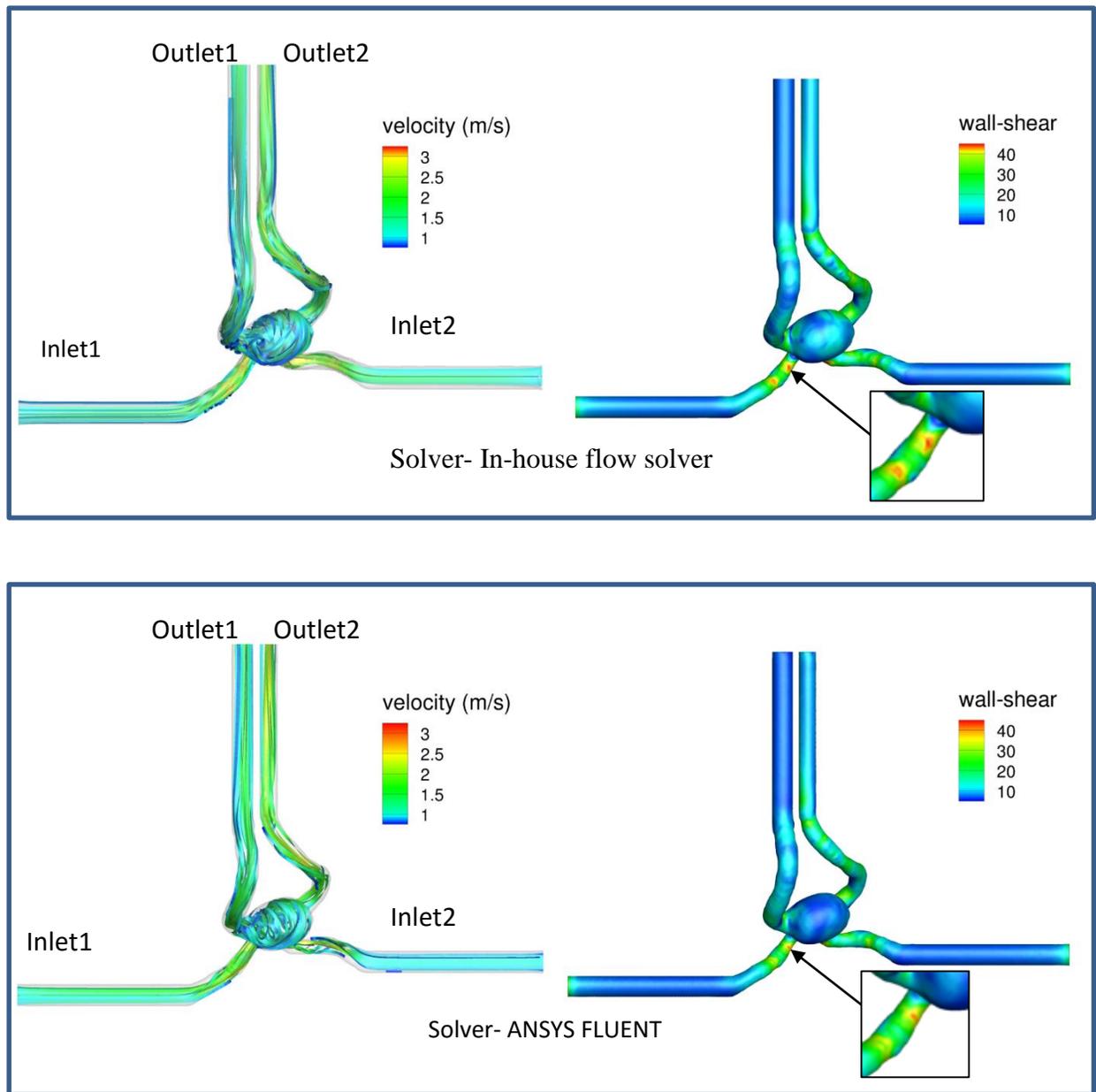


Figure 42 Validation of flow results obtained from image-to-flow set up against flow simulation obtained from FLUENT v14.5.0 (ANSYS Inc., Canonsburg, PA) for intra cranial aneurysm (ICA) for the same boundary conditions. Stream ribbons of velocity vectors colored by velocity magnitude from in-house flow solver (top-left) and ANSYS FLUENT v14.5.0 (ANSYS Inc., Canonsburg, PA) (bottom-left) are shown in this figure. Similarly, corresponding plots of wall shear stress are shown in right side of this figure. Insets point at the regions where results from in house solver appear slightly higher compared to FLUENT v14.5.0 (ANSYS Inc., Canonsburg, PA).

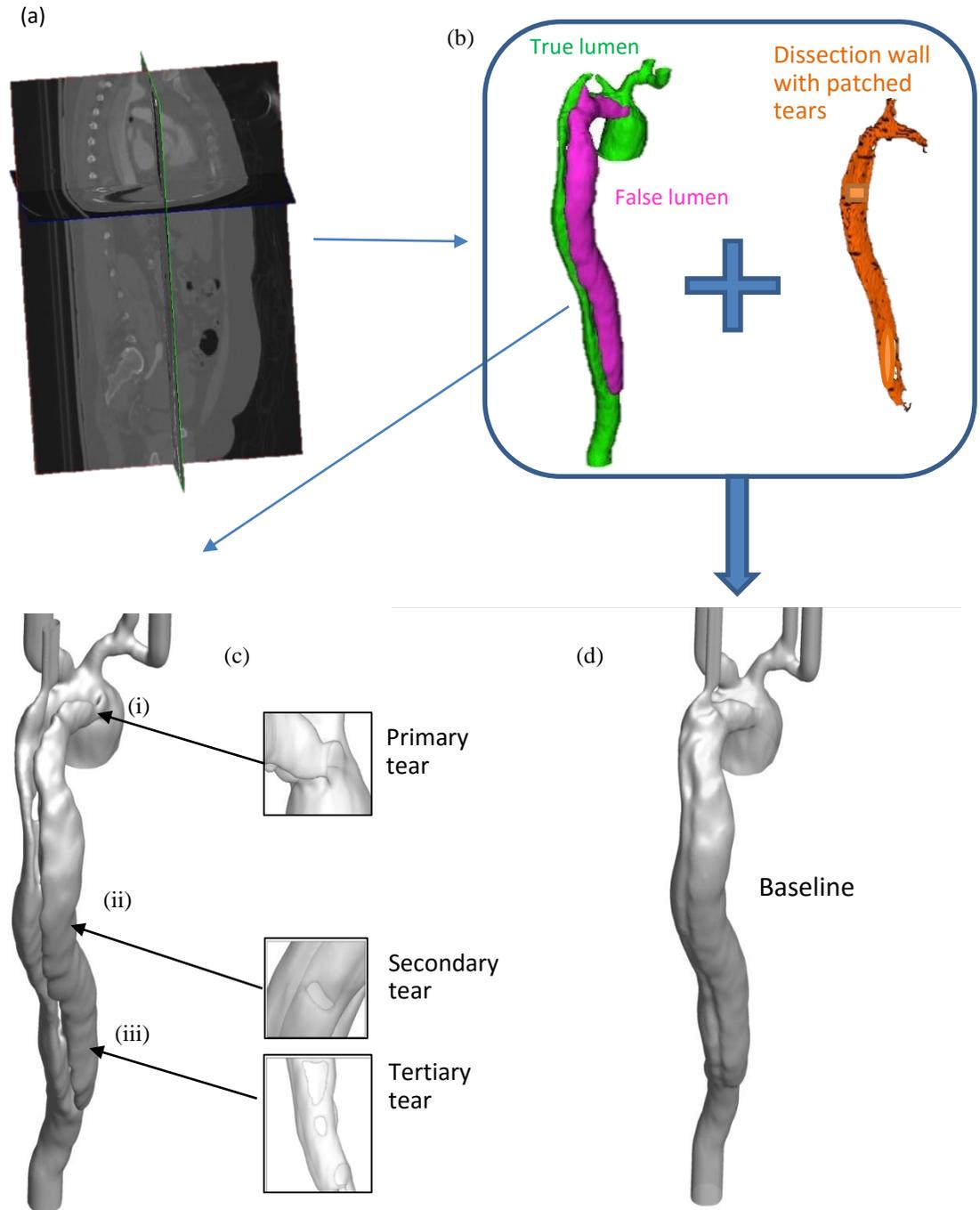


Figure 43 (a) CT scan of a patient with Type-A dissection; (b) visualization of true lumen (green), false lumen (pink) and dissection wall with patched tears of the dissected aorta; (c) reconstructed Diseased model with insets highlighting the location of tears and (d) Baseline aortic model created by merging true lumen, false lumen and dissection wall with tears patched, representing healthy aorta used in Study 1.

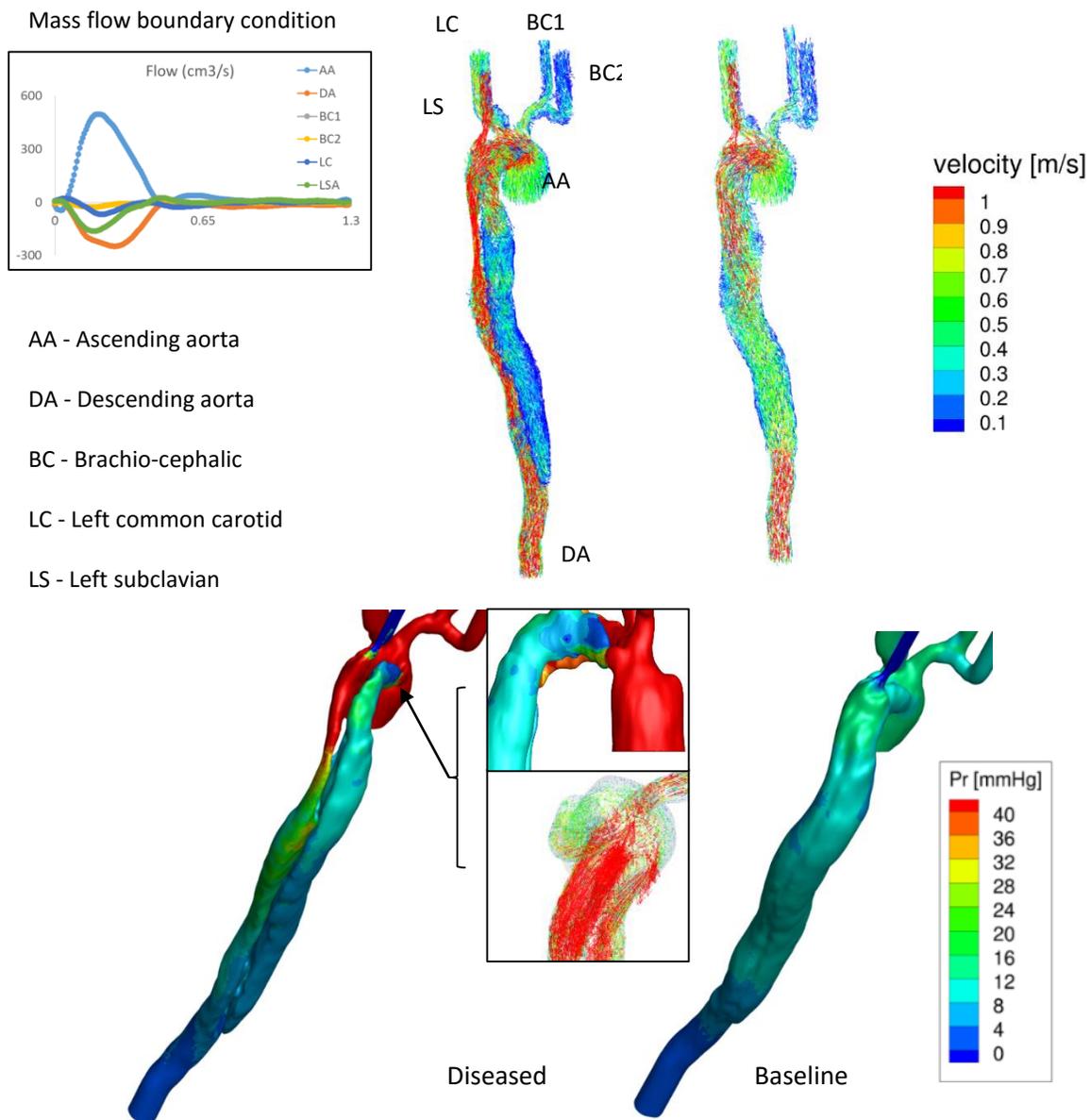


Figure 44 Visualization of velocity (top) and pressure (bottom) in diseased model with Type-A dissection and baseline model representing healthy aorta at peak systole. Boundary conditions applied at different inlets and outlets are shown in left. Pressure at Peak systole for normal aorta model (top-left), diseased before surgery case (top-right), after-surgery case (bottom-left), and after-surgery true lumen expanded case (bottom-right). Total pressure in ascending aorta becomes higher in before-surgery case than in normal model due to the decrease in cross sectional area for the same amount of mass influx.. Since the cross sectional area for mass flow decreases further in after-surgery model due to the closure of entry point of false lumen, the pressure at the ascending aorta escalates further.

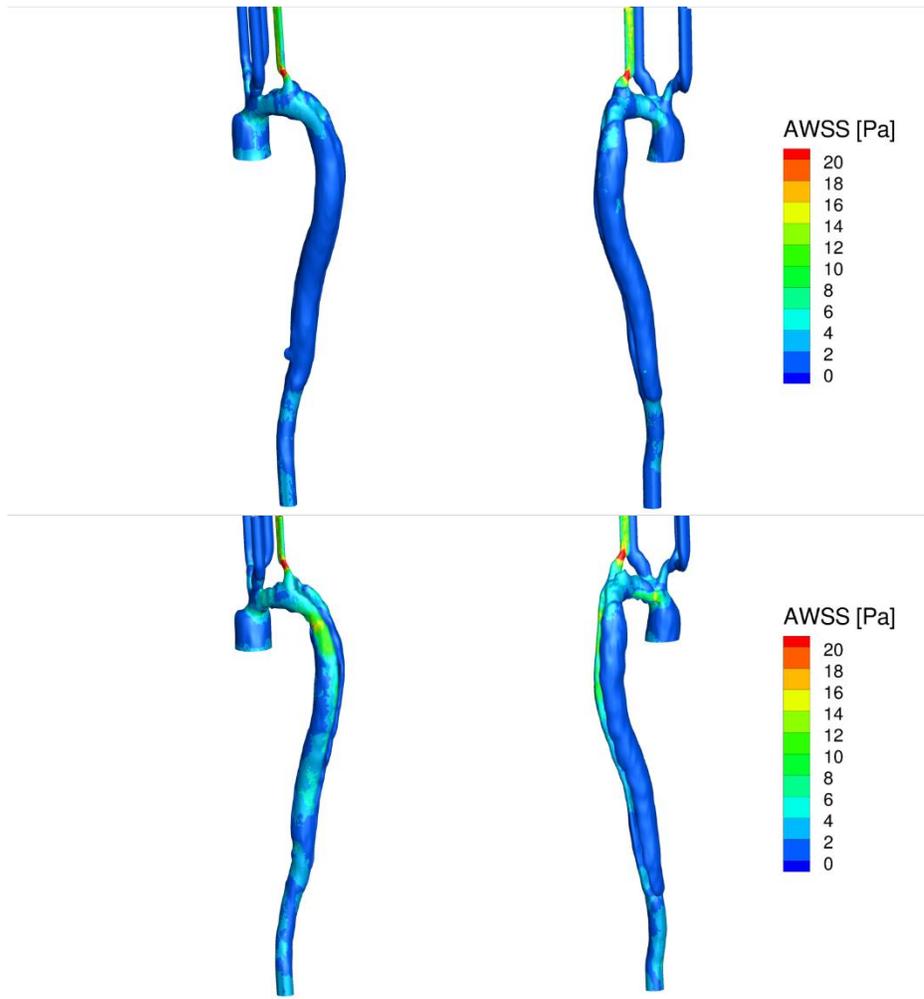


Figure 45 Visualization of Time Averaged Wall Shear Stress (AWSS) for baseline undissected (top) and diseased model (bottom). Comparison of both the views shown in left and right show the presence of higher AWSS in cross sections with minimal cross section (left) and entry point of primary tear (right) in diseased case than in baseline case.

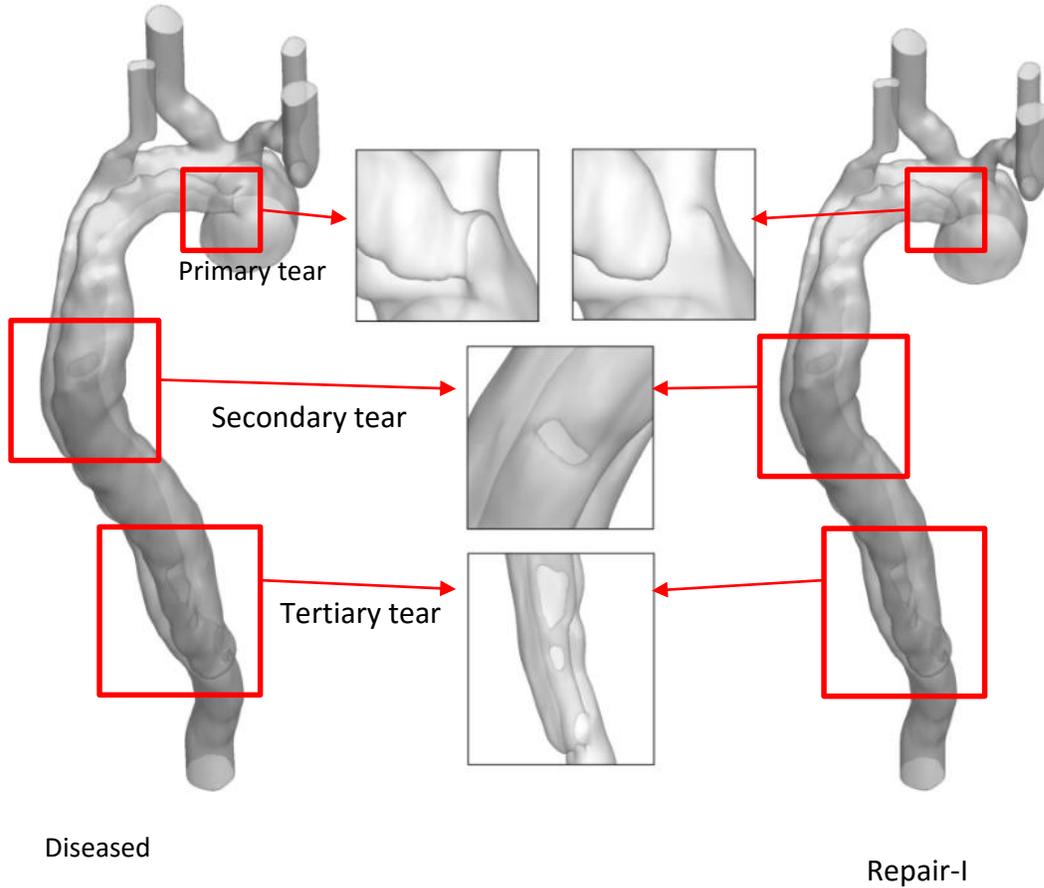


Figure 46 Model of Type-A aortic dissection (left) with 5 tears and Repair-I (right most). The first tear is at the junction of ascending aorta and arch, the location is highlighted in inset in the middle. This tear is removed in the after surgery model as shown in inset. Second tear is at the middle of descending aorta and exists in both before surgery and after surgery models. It's magnified view is shown in inset. Three tears are present at the bottom of descending aorta which exist in both before and after surgery models and the magnified view is shown in the inset in the middle.

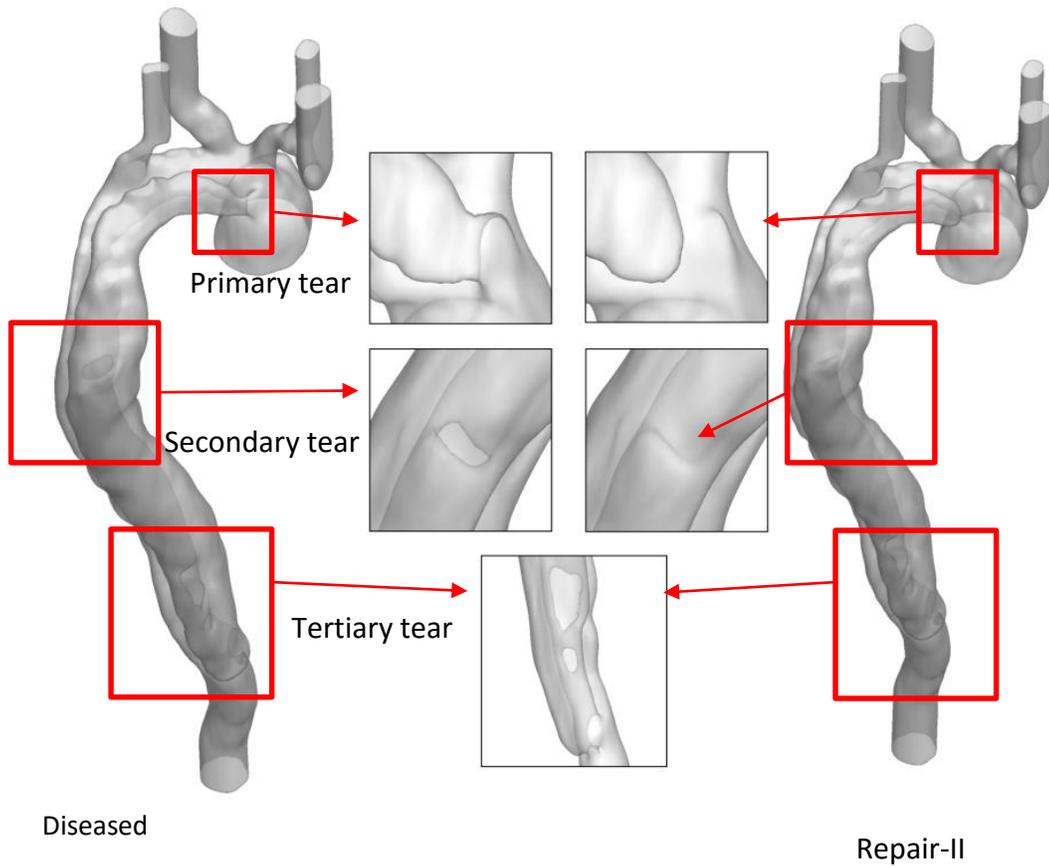


Figure 47 Models of Diseased (left) and Repair-II (right) cases highlighting the differences in the insets. Primary and secondary tears present in original Diseased model are patched in Repair-II model which can be seen in insets in first and second row. Three tertiary tears present in Diseased case are kept intact in Repair-II model as shown in inset in third row.

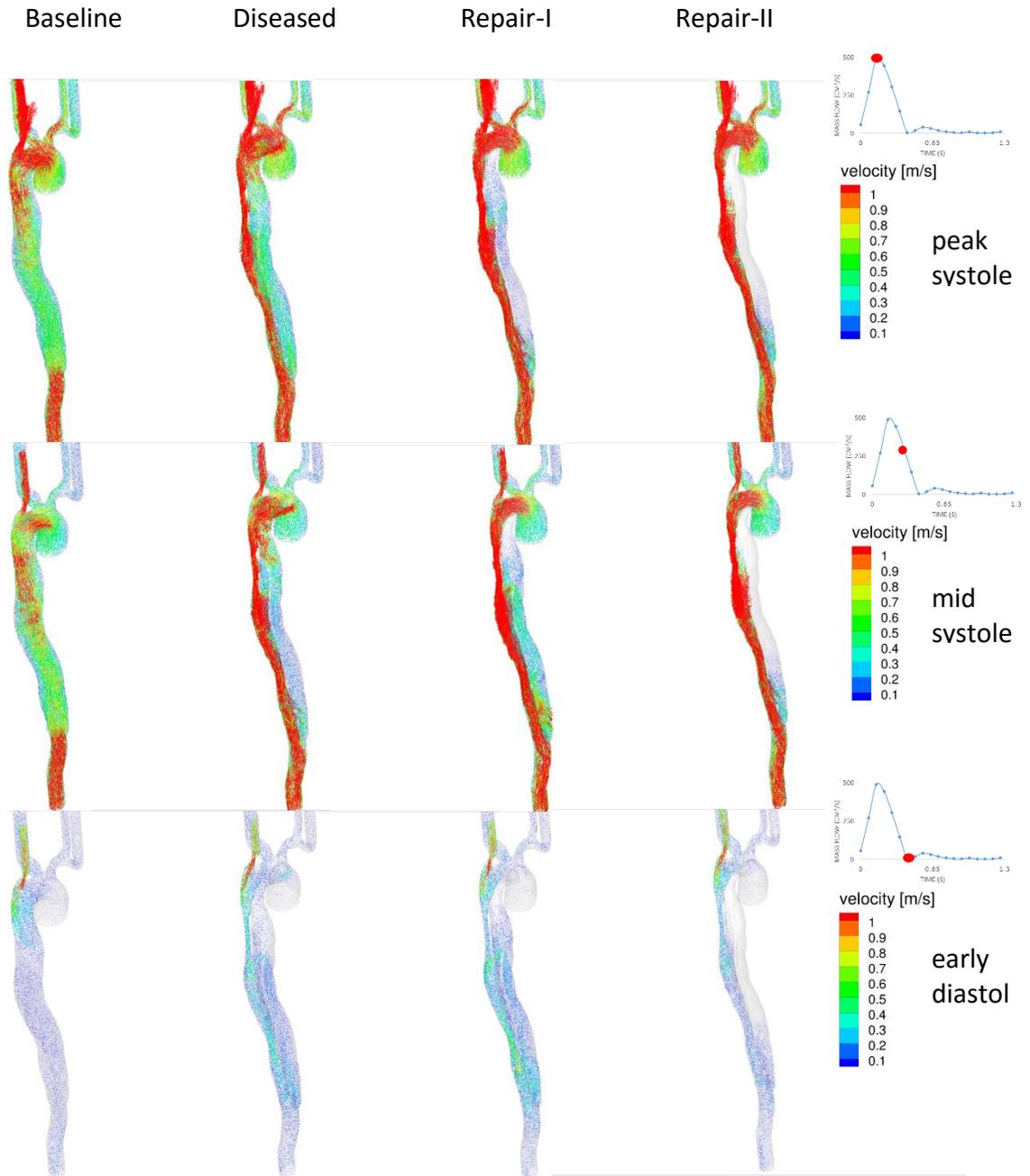


Figure 48 A comparison of velocity vectors of dissection models- baseline, Diseased, Repair-I and Repair-II from left to right, colored and scaled by velocity magnitude at peak systole (top), mid systole (middle) and early diastole (bottom). Exact point in cardiac cycle captured in the images have been highlighted by red dot in the mass flow Vs time plot in the right. Flow in the false lumen decreases throughout the cardiac cycle in both repair cases when compared to Diseased case. However, flow is localized to bottom tertiary tears in Repair-II while in the region extending from the secondary tear to tertiary tears in Repair-I.

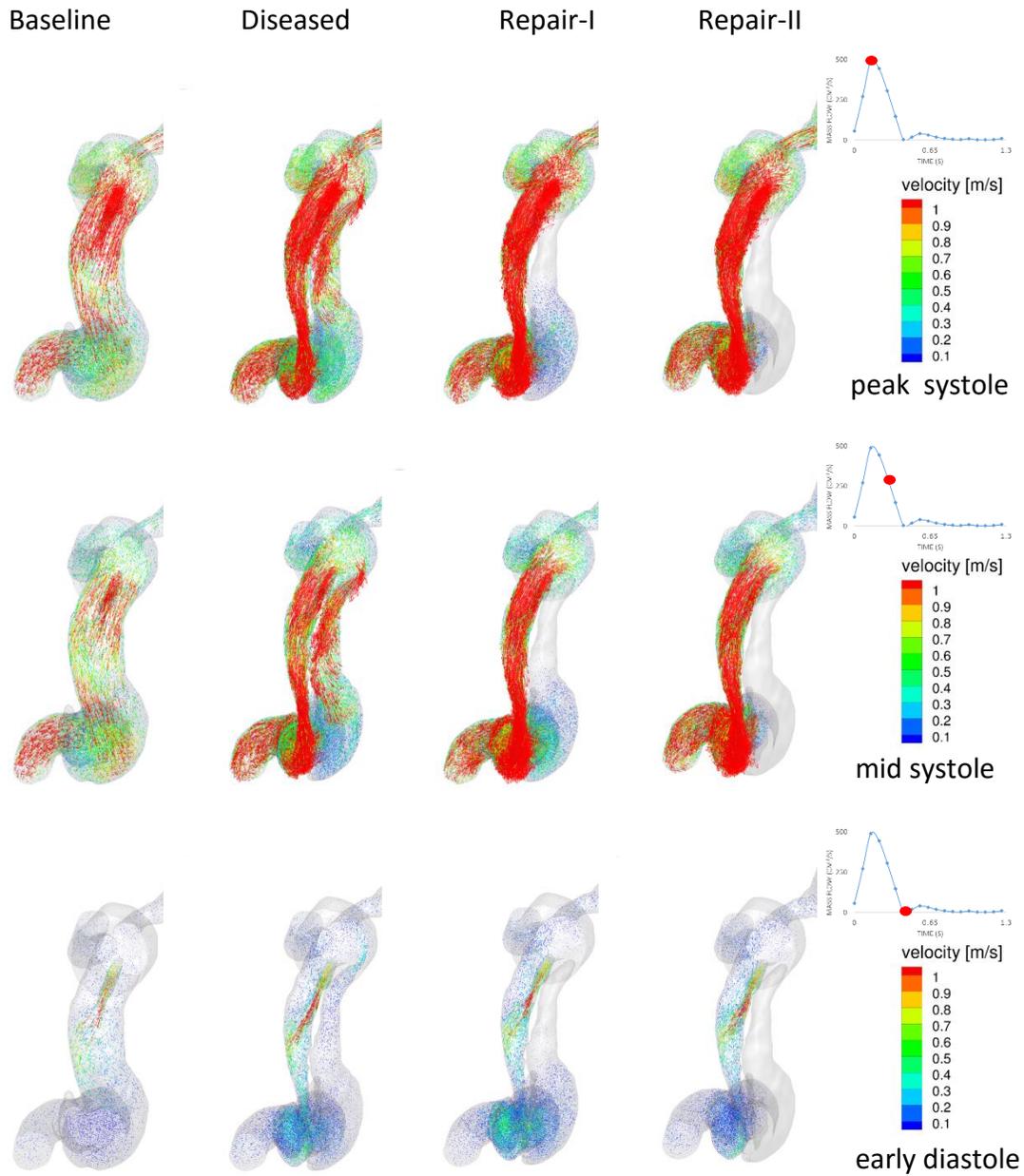


Figure 49 Top view of velocity vectors of dissection models- baseline, Diseased, Repair-I and Repair-II from left to right, colored and scaled by velocity magnitude at peak systole (top), mid systole (middle) and early diastole (bottom). Exact point in cardiac cycle captured in the images have been highlighted by red dot in the mass flow Vs time plot in the right. Flow in the false lumen decreases throughout the cardiac cycle in both repair cases when compared to Diseased case. However, flow is localized to bottom tertiary tears in Repair-II while in the region extending from the secondary tear to tertiary tears in Repair-I.

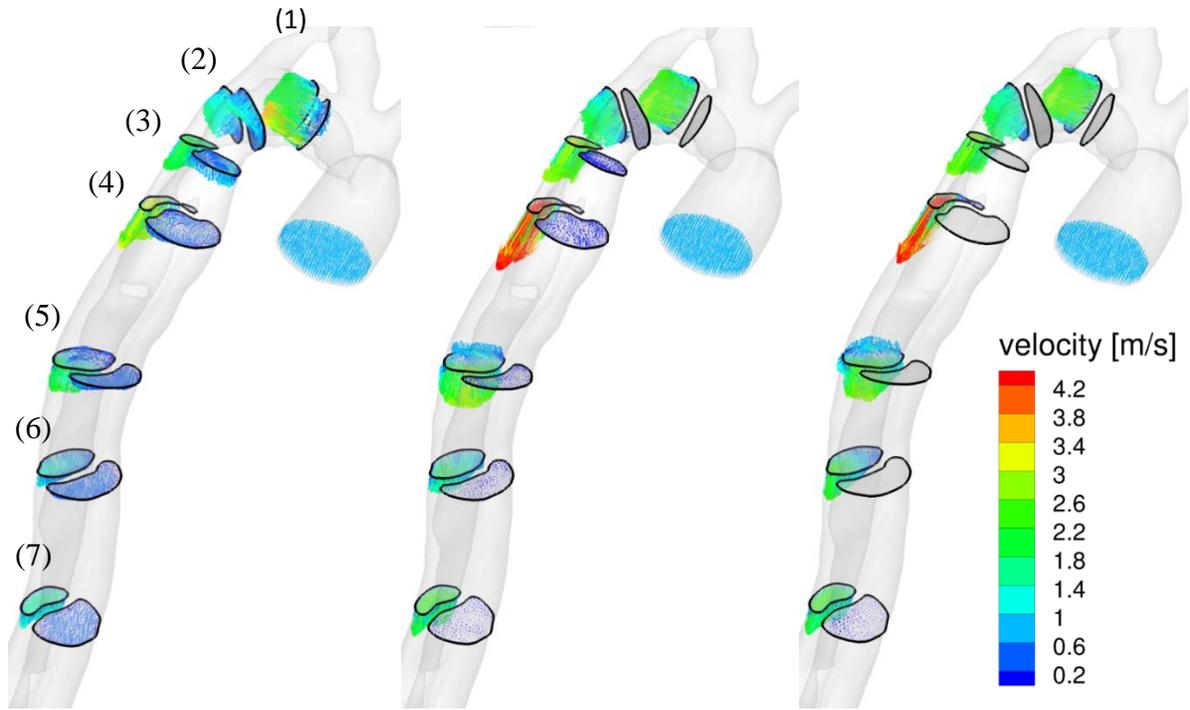


Figure 50 A comparison of velocity vectors at same cross sections for Diseased, Repair-I and Repair-II case to check the effectiveness of Repair-I model in reducing mass flux in false lumen.

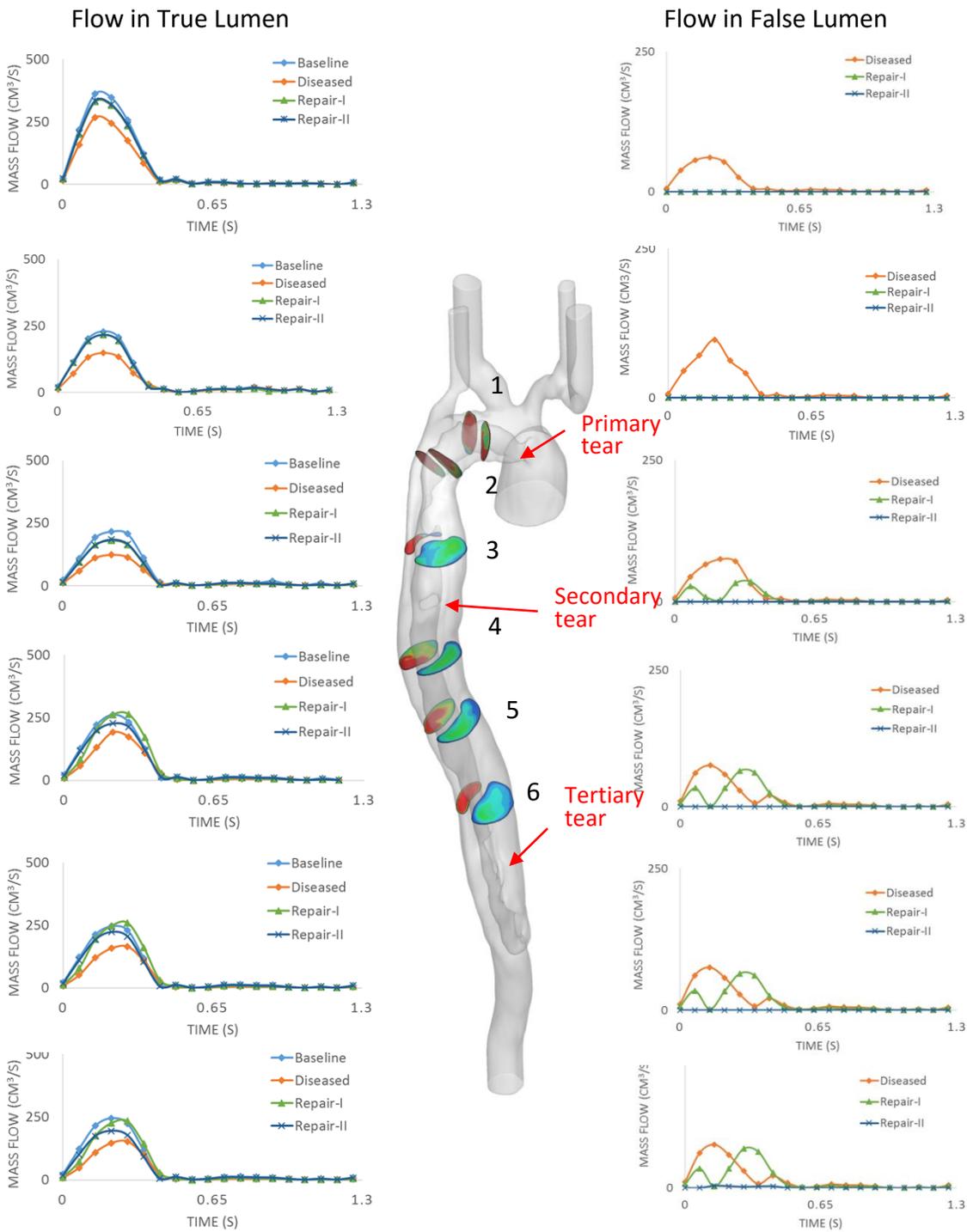


Figure 51 Mass flow rate throughout a cardiac cycle in true lumen (right) and false lumen (left) at different cross sections with respect to time for baseline (true lumen only), Diseased, Repair-I and Repair-II cases. The cross sections analyzed are shown in the middle of the figure.

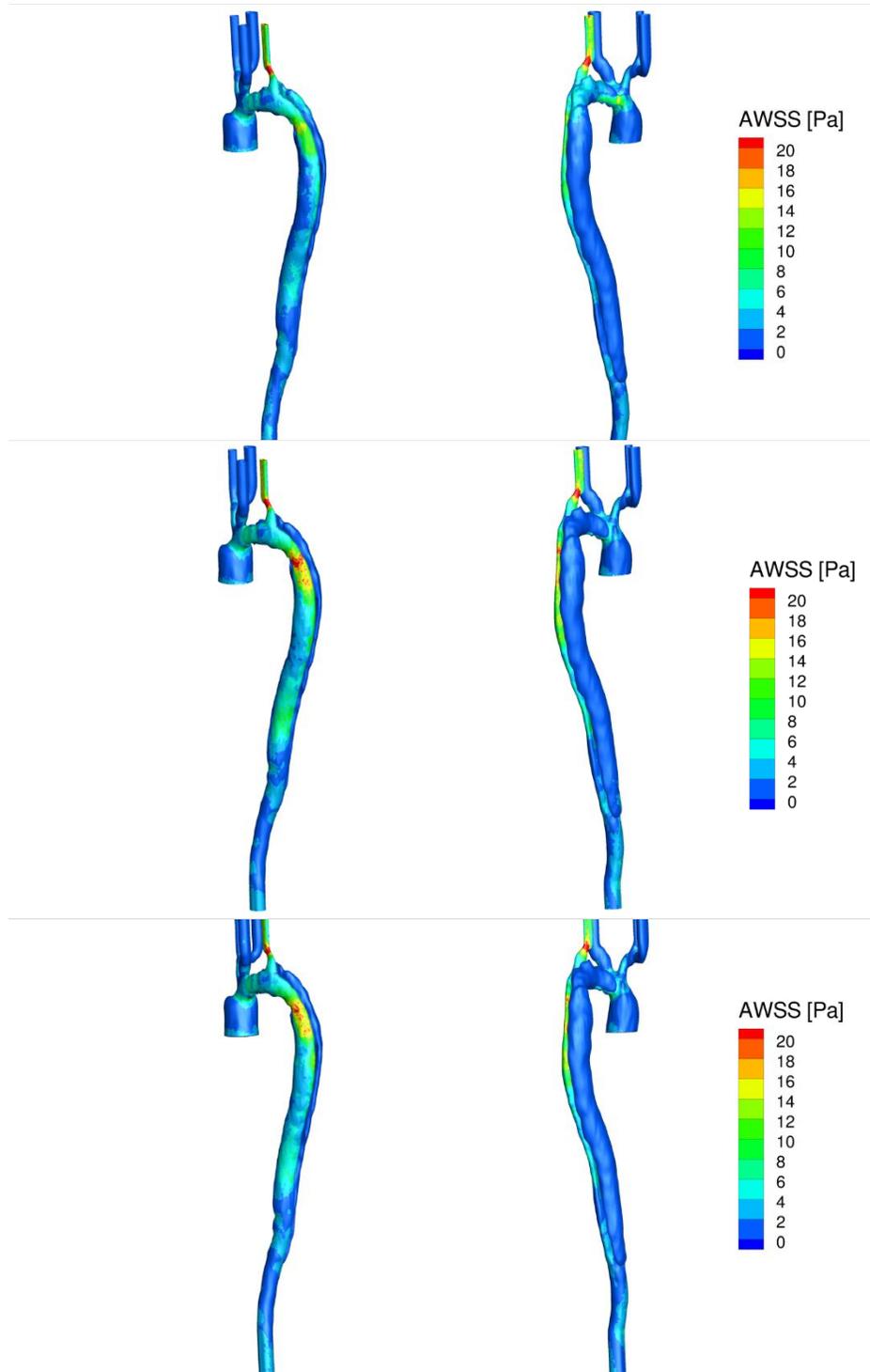


Figure 52 A comparison of time averaged wall shear stress (AWSS) for diseased (top), repair-I (middle) and repair-II (bottom). Views of true lumen and false lumen of the same model are shown in left and right. Higher AWSS is seen in true lumen mainly in the collapsed region as the flow is restored after repair in the two repair models. Similarly, low AWSS is seen in the false lumen at the arch region due to the patching of primary tear in the repair models.

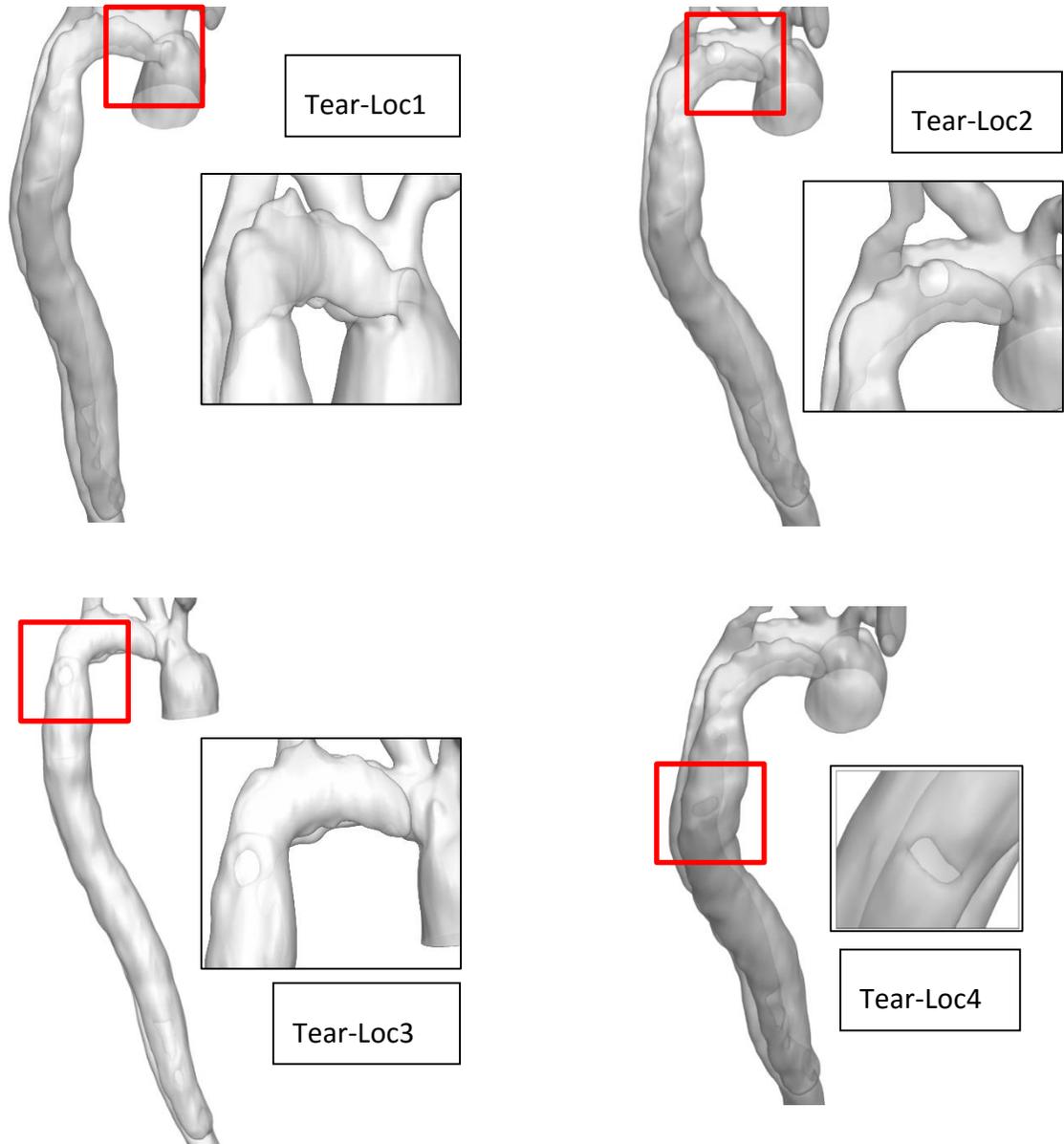


Figure 53 Models of diseased cases where the location of primary tear are altered to understand the role of location of tear in the mass flow in aortic arch. Model with primary tear located at ascending aorta (top-left), model with primary tear located before left-subclavian artery in arch region (top-right), model with primary tear located in descending aorta after left left-subclavian artery (bottom-left), model with primary tear located at the middle of descending aorta (bottom-right) are shown in this figure and they are addressed as Tear-Loc1, Tear-Loc2, Tear-Loc3, and Tear-Loc4 (or Repair-I) respectively.

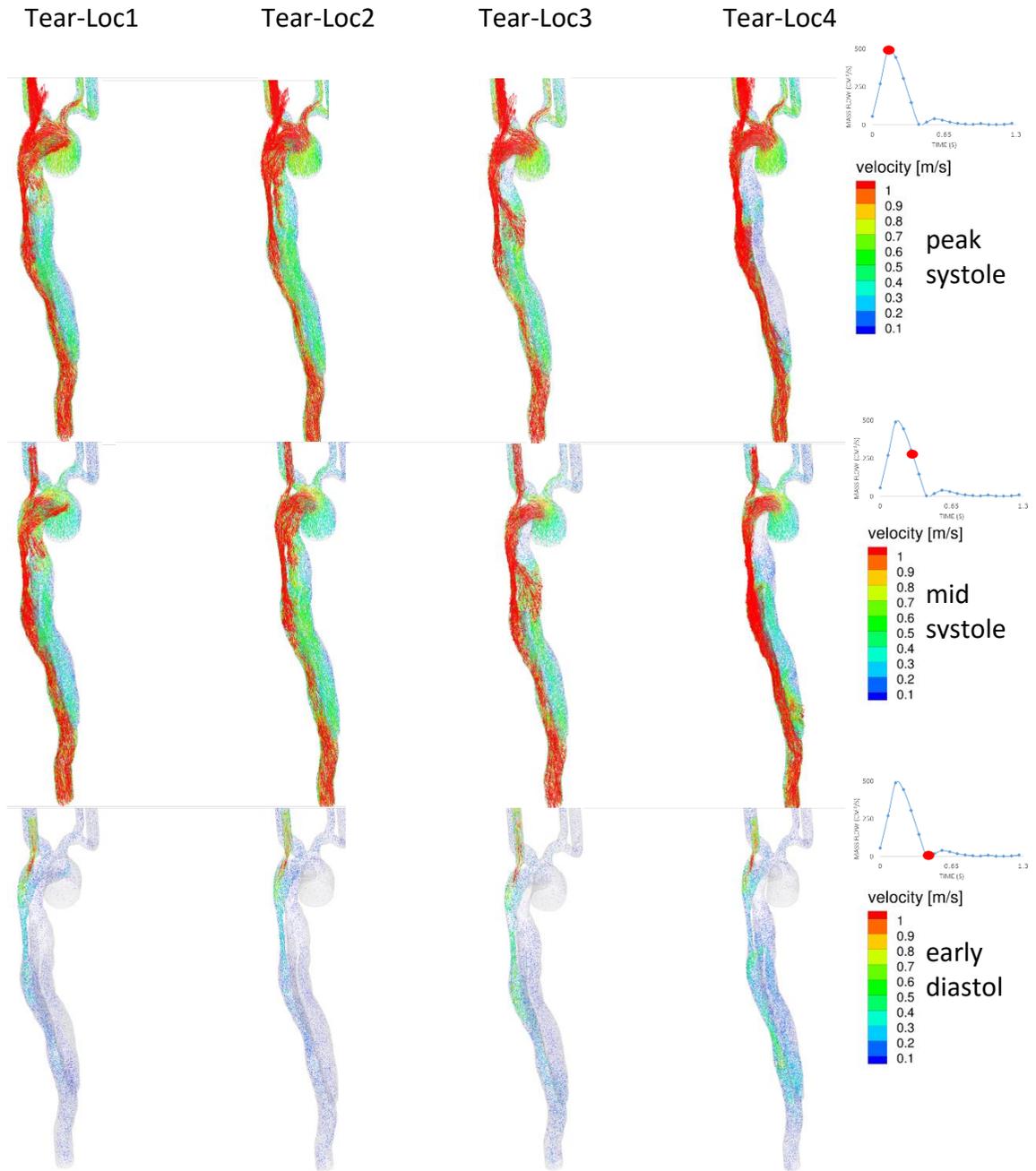


Figure 54 Visualization of velocity vectors at peak systole (top row), mid systole (middle row), and early diastole (bottom row) for dissected models with varied location of primary tear namely, TearLoc1, TearLoc2, TearLoc3, and TearLoc4.

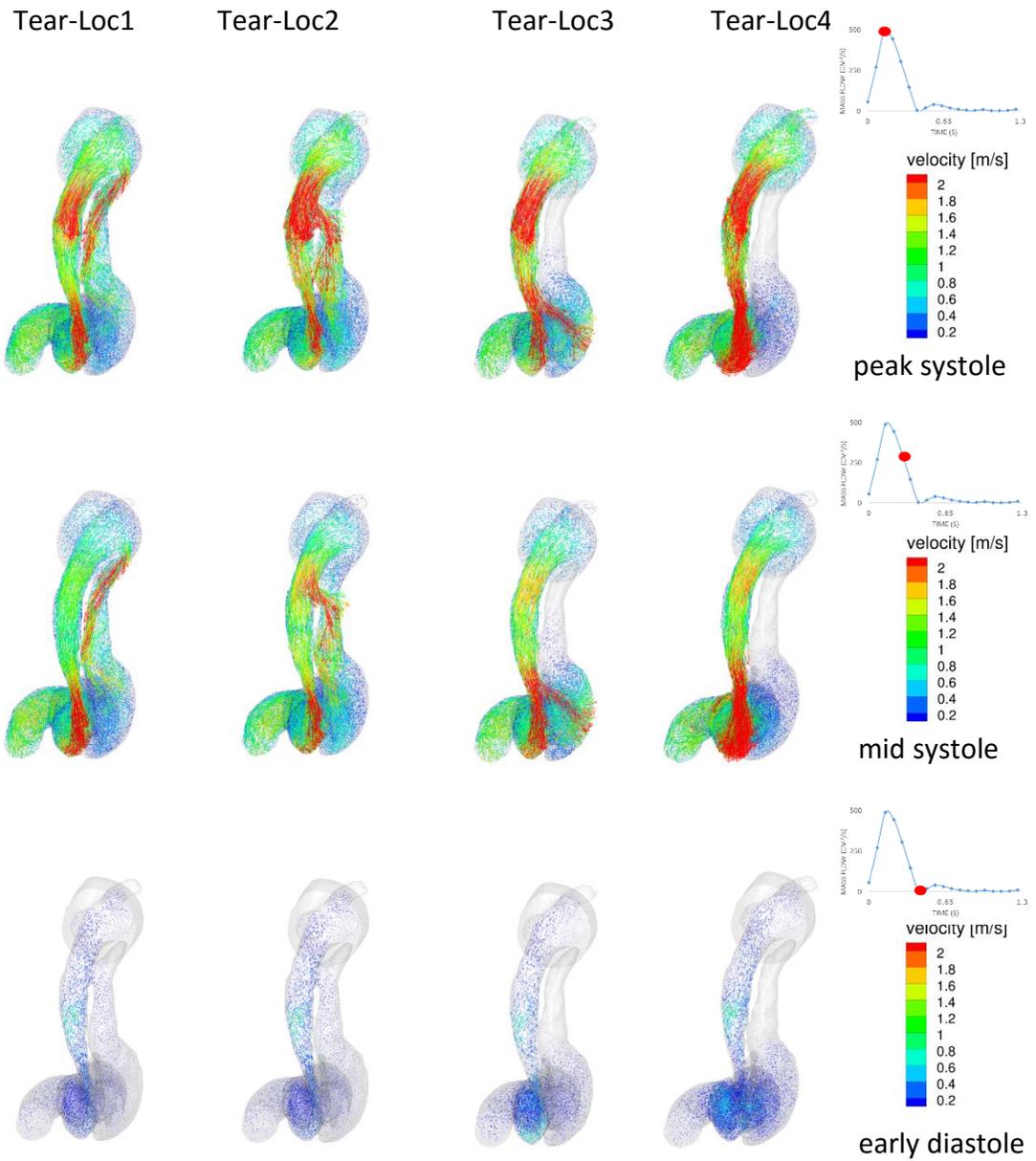


Figure 55 Top view of velocity vectors at peak systole (top row), mid systole (middle row), and early diastole (bottom row) for dissected models with varied location of primary tear namely, TearLoc1, TearLoc2, TearLoc3, and TearLoc4.

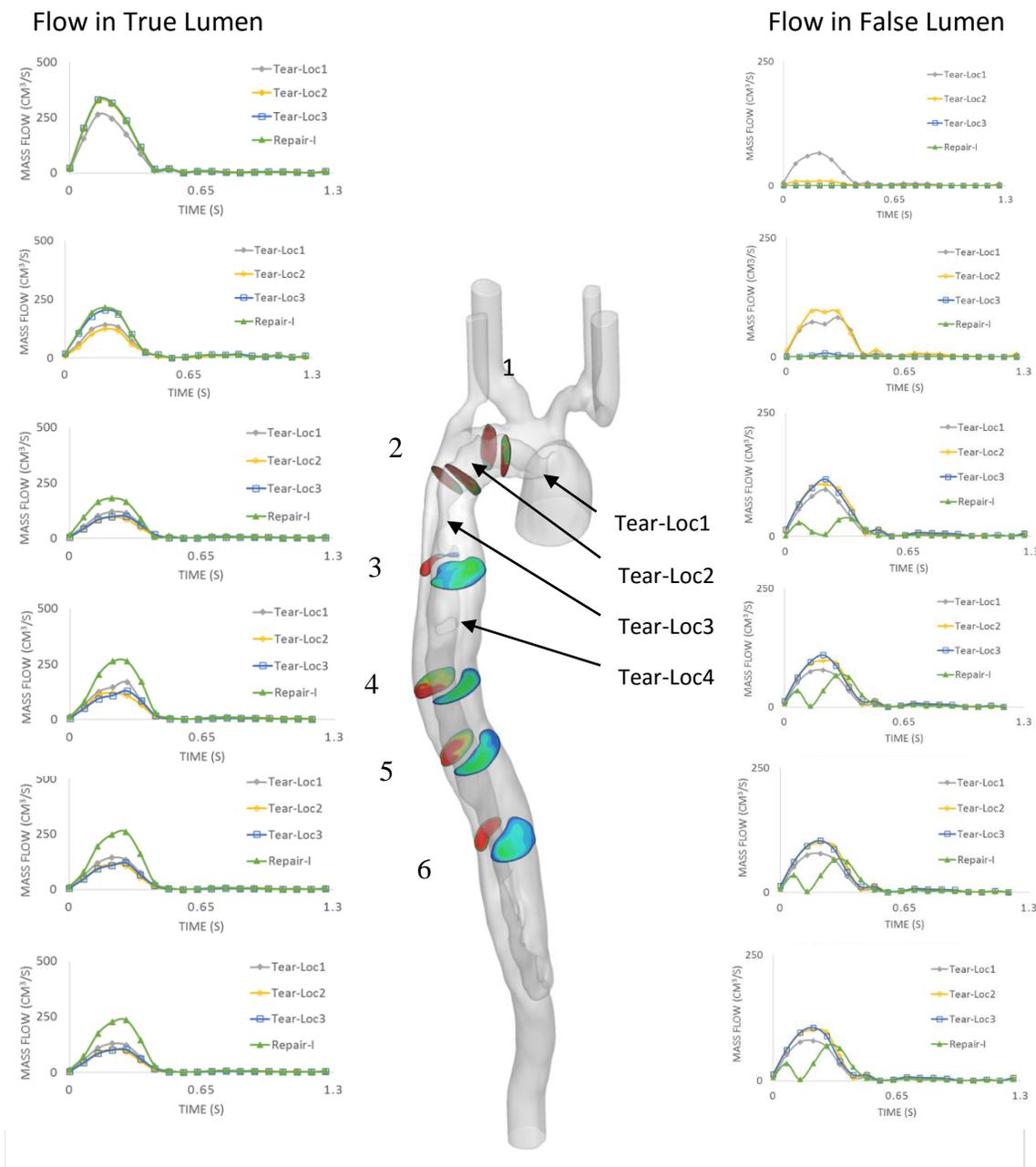


Figure 56 Mass flow rate in true lumen (right) and false lumen (left) at different cross sections with respect to time for models of dissected cases-with primary tear at varying locations, namely, Tear-Loc1, Tear-Loc2, Tear-Loc3, and Tear-Loc4 (or Repair-I). The cross sections analyzed are shown in the middle of the figure.

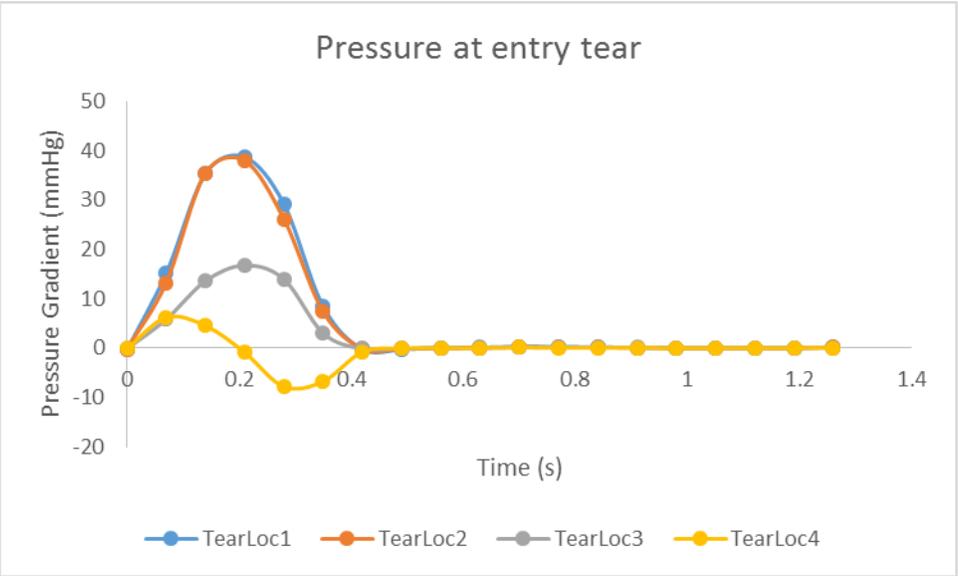


Figure 57 Variation of pressure in primary tear in a cardiac cycle for dissection models with varying location of primary tear, namely, Tear-Loc1, Tear-Loc2, Tear-Loc3, and Tear-Loc4.

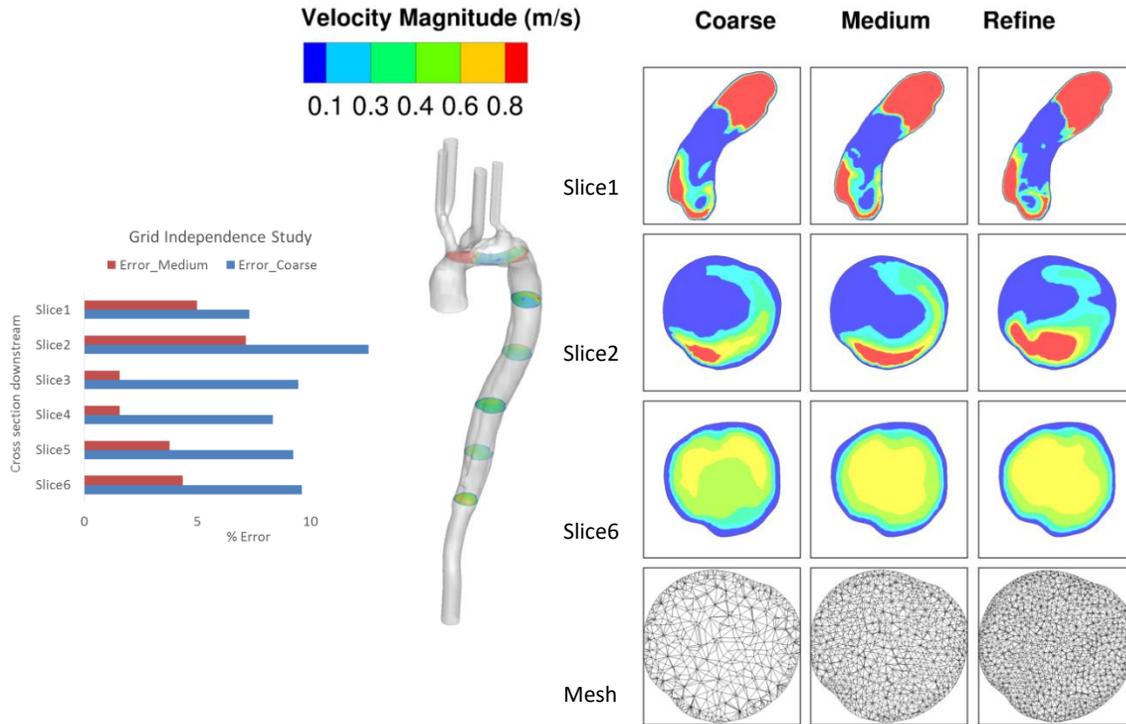


Figure 58 Grid independence study was carried out for normal undissected aorta where three models coarse, medium and refine were generated with average element of size 0.002 mm, 0.001 mm and 0.00075 mm respectively. Velocity magnitudes for all the models at six cross sections as shown (middle) were analyzed. Three of the six cross sections are shown as slice1, slice 2 and slice 3 on right. Percentage error (left) were calculated for these cross sections with respect to the refine case. Medium case with average element size of 0.001 mm was selected for this study.

BIBLIOGRAPHY

- A. A. Amini, T. E. Weymouth. 1990. "Using Dynamic Programming for Solving Variational Problems Invision." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 12 (9): 855–67. doi:10.1109/34.57681.
- Alimohammadi, Mona, Obiekezie Agu, Stavroula Balabani, and Vanessa Díaz-Zuccarini. 2016. "Development of a Patient-Specific Simulation Tool to Analyse Aortic Dissections: Assessment of Mixed Patient-Specific Flow and Pressure Boundary Conditions." *Medical Engineering and Physics* 36 (3): 275–84. Accessed January 30. doi:10.1016/j.medengphy.2013.11.003.
- Angelini, Elsa, Yinpeng Jin, and Andrew Laine. 2004. "State of the Art of Level Set Methods in Segmentation and Registration of Medical Imaging Modalities." In *Handbook of Biomedical Image Analysis*, edited by Jasjit S. Suri, David L. Wilson, and Swamy Laxminarayan, 47–101. Boston, MA: Springer US. <http://academiccommons.columbia.edu/catalog/ac:128463>.
- Antiga, L. 2002. "Patient-Specific Modeling of Geometry and Blood Flow in Large Arteries." Politecnico di Milano.
- Antiga, Luca, Marina Piccinelli, Lorenzo Botti, Bogdan Ene-Iordache, Andrea Remuzzi, and David A Steinman. 2008. "An Image-Based Modeling Framework for Patient-Specific Computational Hemodynamics." *Medical & Biological Engineering & Computing* 46 (11): 1097–1112. doi:10.1007/s11517-008-0420-1.
- Antiga, Luca, and David A Steinman. 2006. "VMTK-Vascular Modeling Toolkit." <http://www.vmtk.org>.
- Bagchi, Prosenjit, Paul C Johnson, and Aleksander S Popel. 2005. "Computational Fluid Dynamic Simulation of Aggregation of Deformable Cells in a Shear Flow." *Journal of Biomechanical Engineering* 127 (7): 1070–80.
- Beek, A. J. van. 1997. "A Finite Element Model of the Urinary Bladder." <http://alexandria.tue.nl/repository/books/491375.pdf>.
- Beucher, S, X Yu, and M Bileadu. 1990. "Road Segmentataion by Watershed Algorithm." In *Workshop in Vision*. Sophia AntipolisFrance.
- Burstedde, Carsten, Lucas C. Wilcox, and Omar Ghattas. 2011. "*p4est*: Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees." *SIAM J. Sci. Comput.* 33 (3): 1103–33. doi:10.1137/100791634.
- Caselles, Vicent, Ron Kimmel, and Guillermo Sapiro. 1995. "Geodesic Active Contours." *International Journal of Computer Vision* 22: 61–79.
- Castro, M. A., C. M. Putman, and J. R. Cebal. 2006. "Computational Fluid Dynamics Modeling of Intracranial Aneurysms: Effects of Parent Artery Segmentation on Intra-Aneurysmal Hemodynamics." *American Journal of Neuroradiology* 27 (8): 1703–9. <http://www.ajnr.org/content/27/8/1703>.
- Cebal, J. R., F. Mut, D. Sforza, R. Löhner, E. Scrivano, P. Lylyk, and C. Putman. 2011. "Clinical Application of Image-Based CFD for Cerebral Aneurysms." *International Journal for Numerical Methods in Biomedical Engineering* 27 (7): 977–92. doi:10.1002/cnm.1373.
- Cebal, JR, F Mut, D Sforza, R Löhner, E Scrivano, P Lylyk, and CM Putman. 2011. "Clinical Application of Image-Based CFD for Cerebral Aneurysms." *International Journal for Numerical Methods in Biomedical Engineering* 27 (7): 977–92. doi:10.1002/cnm.1373.

- Cebral, Juan R, Marcelo A Castro, Sunil Appanaboyina, Christopher M Putman, Daniel Millan, and Alejandro F Frangi. 2005. "Efficient Pipeline for Image-Based Patient-Specific Analysis of Cerebral Aneurysm Hemodynamics: Technique and Sensitivity." *IEEE Transactions on Medical Imaging* 24 (4): 457–67.
- Cebral, Juan R., Monica Hernandez, Alejandro Frangi, Christopher Putman, Richard Pergolizzi, and James Burgess. 2004. "Subject-Specific Modeling of Intracranial Aneurysms." In , 5369:319–27. doi:10.1117/12.535441.
- Cezeaux, Judy L., and Albertus van Grondelle. 1997. "Accuracy of the Inverse Womersley Method for the Calculation of Hemodynamic Variables." *Annals of Biomedical Engineering* 25 (3): 536–46. doi:10.1007/BF02684193.
- Chandran, Krishnan B., Ajit Prithiviraj Yoganathan, and Stanley E. Rittgers. 2012. *Biofluid Mechanics: The Human Circulation*. CRC Press.
- Chan, T.F., and L.A. Vese. 2001. "Active Contours without Edges." *IEEE Transactions on Image Processing* 10 (2): 266–77. doi:10.1109/83.902291.
- Chen, Duanduan, Matthias Müller-Eschner, Hendrik von Tengg-Kobligk, David Barber, Dittmar Böckler, Rod Hose, and Yiannis Ventikos. 2013. "A Patient-Specific Study of Type-B Aortic Dissection: Evaluation of True-False Lumen Blood Exchange." *BioMedical Engineering OnLine* 12 (1): 65. doi:10.1186/1475-925X-12-65.
- Chessa, J., and T. Belytschko. 2003. "An Extended Finite Element Method for Two-Phase Fluids." *Journal of Applied Mechanics* 70 (1): 10–17. doi:10.1115/1.1526599.
- Chisari, N. G., G. Artana, and D. Sciamarella. 2011. "Vortex Dipolar Structures in a Rigid Model of the Larynx at Flow Onset - Springer." *Experiments in Fluids* 50 (2): 397–406.
- Dillard, Seth I., John A. Mousel, Liza Shrestha, Madhavan L. Raghavan, and Sarah C. Vigmostad. 2014. "From Medical Images to Flow Computations without User-Generated Meshes." *International Journal for Numerical Methods in Biomedical Engineering* 30 (10): 1057–83. doi:10.1002/cnm.2644.
- Dillard, Seth, Sreedevi Krishnan, and Udaykumar H S. 2007. "Mechanics of Flow and Mixing at Antroduodenal Junction." *World Journal of Gastroenterology : WJG* 13 (9): 1365–71.
- Dillon-Murphy, Desmond, Alia Noorani, David Nordsletten, and C. Alberto Figueroa. 2015. "Multi-Modality Image-Based Computational Analysis of Haemodynamics in Aortic Dissection." *Biomechanics and Modeling in Mechanobiology*, September, 1–20. doi:10.1007/s10237-015-0729-2.
- "Extreme Programming." 2013. *Extreme Programming: A Gentle Introduction*. <http://www.extremeprogramming.org/>.
- Gefen, Amit. 2012. *Patient-Specific Modeling in Tomorrow's Medicine*. Springer.
- Glasbey, C. A., and G. W. Horgan. 1995. *Image Analysis for the Biological Sciences*. New York, NY, USA: John Wiley & Sons, Inc.
- Grau, V, A U J Mewes, M Alcañiz, R Kikinis, and S K Warfield. 2004. "Improved Watershed Transform for Medical Image Segmentation Using Prior Information." *IEEE Transactions on Medical Imaging* 23 (4): 447–58. doi:10.1109/TMI.2004.824224.
- Grosland, Nicole M., Kiran H. Shivanna, Vincent A. Magnotta, Nicole A. Kallemeyn, Nicole A. DeVries, Srinivas C. Tadepalli, and Curtis Lisle. 2009. "IA-FEMesh: An Open-Source, Interactive, Multiblock Approach to Anatomic Finite Element Model Development." *Computer Methods and Programs in Biomedicine* 94 (1): 96–107. doi:10.1016/j.cmpb.2008.12.003.

- György Paál, Ádám Ugron. 2007. “Flow in Simplified and Real Models of Intracranial Aneurysms.” *International Journal of Heat and Fluid Flow*, no. 4 (August): 653–64. doi:10.1016/j.ijheatfluidflow.2007.04.004.
- Hegadi, Ravindra, Arpana Kop, and Mallikarjun Hangarge. 2010. “A Survey on Deformable Model and Its Applications to Medical Imaging.” *International Journal of Computer Applications IJCA RTIPPR* (2): 64–75.
- Hernandez, Monica, Rosario Barrena, Gabriel Hernandez, Guillermo Sapiro, and Alejandro F. Frangi. 2003. “Pre-Clinical Evaluation of Implicit Deformable Models for Three-Dimensional Segmentation of Brain Aneurysms from CTA Images.” In , 5032:1264–74. doi:10.1117/12.483596.
- Hoh, Brian L., Christopher L. Siström, Christopher S. Firment, Gregory L. Fautheree, Gregory J. Velat, Jobyna H. Whiting, John F. Reavey-Cantwell, and Stephen B. Lewis. 2007. “Bottleneck Factor and Height-Width Ratio: Association with Ruptured Aneurysms in Patients with Multiple Cerebral Aneurysms.” *Neurosurgery* 61 (4): 716–22; discussion 722–23. doi:10.1227/01.NEU.0000298899.77097.BF.
- Johnson, T. A., and V. C. Patel. 1999. “Flow Past a Sphere up to a Reynolds Number of 300.” *Journal of Fluid Mechanics* 378: 19–70. doi:10.1017/S0022112098003206.
- Karmonik, C., J. Bismuth, D. J. Shah, M. G. Davies, D. Purdy, and A. B. Lumsden. 2011. “Computational Study of Haemodynamic Effects of Entry- and Exit-Tear Coverage in a DeBakey Type III Aortic Dissection: Technical Report.” *European Journal of Vascular and Endovascular Surgery* 42 (2): 172–77. doi:10.1016/j.ejvs.2011.04.008.
- Karmonik, Christof, Jean Bismuth, Mark G. Davies, Dipan J. Shah, Houssam K. Younes, and Alan B. Lumsden. 2011. “A Computational Fluid Dynamics Study Pre- and Post-Stent Graft Placement in an Acute Type B Aortic Dissection.” *Vascular and Endovascular Surgery* 45 (2): 157–64. doi:10.1177/1538574410389342.
- Karmonik, Christof, Matthias Müller-Eschner, Sasan Partovi, Philipp Geisbüsch, Maria-Katharina Ganten, Jean Bismuth, Mark G. Davies, et al. 2013. “Computational Fluid Dynamics Investigation of Chronic Aortic Dissection Hemodynamics versus Normal Aorta.” *Vascular and Endovascular Surgery* 47 (8): 625–31. doi:10.1177/1538574413503561.
- Karmonik, Christof, Sasan Partovi, Matthias Müller-Eschner, Jean Bismuth, Mark G. Davies, Dipan J. Shah, Matthias Loebe, Dittmar Böckler, Alan B. Lumsden, and Hendrik von Tengg-Kobligk. 2012. “Longitudinal Computational Fluid Dynamics Study of Aneurysmal Dilatation in a Chronic DeBakey Type III Aortic Dissection.” *Journal of Vascular Surgery* 56 (1): 260–63.e1. doi:10.1016/j.jvs.2012.02.064.
- Karmonik, Prof Christof, and Dr Gavin W. Britz. 2014. “Computational Fluid Dynamics and Cerebral Aneurysms.” In *PanVascular Medicine*, edited by Peter Lanzer, 1–13. Springer Berlin Heidelberg. http://link.springer.com/referenceworkentry/10.1007/978-3-642-37393-0_33-1.
- Koumoutsakos, P., and A. Leonard. 1995. “High-Resolution Simulations of the Flow around an Impulsively Started Cylinder Using Vortex Methods.” *Journal of Fluid Mechanics* 296: 1–38. doi:10.1017/S0022112095002059.

- Kovács, Tamás, Philippe Cattin, Hatem Alkadhi, Simon Wildermuth, and Gábor Székely. 2006. “Automatic Segmentation of the Aortic Dissection Membrane from 3D CTA Images.” In *Medical Imaging and Augmented Reality*, edited by Guang-Zhong Yang, TianZi Jiang, Dinggang Shen, Lixu Gu, and Jie Yang, 317–24. Lecture Notes in Computer Science 4091. Springer Berlin Heidelberg. http://link.springer.com/chapter/10.1007/11812715_40.
- Krissian, Karl, Jose M. Carreira, Julio Esclarin, and Manuel Maynar. 2014. “Semi-Automatic Segmentation and Detection of Aorta Dissection Wall in MDCT Angiography.” *Medical Image Analysis* 18 (1): 83–102. doi:10.1016/j.media.2013.09.004.
- Kute, Stephanie M., and David A. Vorp. 2001. “The Effect of Proximal Artery Flow on the Hemodynamics at the Distal Anastomosis of a Vascular Bypass Graft: Computational Study.” *Journal of Biomechanical Engineering* 123 (3): 277–83. doi:10.1115/1.1374203.
- Lai, Ming-Chih, and Charles S. Peskin. 2000. “An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity.” *Journal of Computational Physics* 160 (2): 705–19. doi:10.1006/jcph.2000.6483.
- Leveque, Randall J., and Zhilin Li. 1994. “The Immersed Interface Method for Elliptic Equations with Discontinuous Coefficients and Singular Sources.” *SIAM J. Num. Anal.*, 1019–44.
- Lin, Ching-Long, Merryn H Tawhai, Geoffrey McLennan, and Eric A Hoffman. 2007. “Characteristics of the Turbulent Laryngeal Jet and Its Effect on Airflow in the Human Intra-Thoracic Airways.” *Respiratory Physiology & Neurobiology* 157 (2-3): 295–309. doi:10.1016/j.resp.2007.02.006.
- Liza Shrestha, Jennifer Burken. 2012. “Hemodynamics in the Aortic Arch and the Relationship to Rupture During Type-A Dissection.”
- Li, Zhilin, and Ming-Chih Lai. 2001. “The Immersed Interface Method for the Navier-Stokes Equations with Singular Forces.” *Journal of Computational Physics*, 2001.
- Lorenson, William E., and Harvey E. Cline. 1987. “Marching Cubes: A High Resolution 3D Surface Construction Algorithm.” *SIGGRAPH Comput. Graph.* 21 (4): 163–69. doi:10.1145/37402.37422.
- Luis, Ibanez, Schroeder Will, Ng Lydia, Cates Josh, and Insight Software Consortium. 2005. *The ITK Software Guide*.
- Luo, X.Y., J.S. Hinton, T.T. Liew, and K.K. Tan. 2004. “LES Modeling of Flow in a Simple Airway Model.” *Medical Engineering & Physics* 26 (5): 403–13. doi:10.1016/j.medengphy.2004.02.008.
- Ma, Baoshun, Robert E. Harbaugh, and Madhavan L. Raghavan. 2004. “Three-Dimensional Geometrical Characterization of Cerebral Aneurysms.” *Annals of Biomedical Engineering* 32 (2): 264–73. doi:10.1023/B:ABME.0000012746.31343.92.
- Malladi, R., J.A. Sethian, and B.C. Vemuri. 1995. “Shape Modeling with Front Propagation: A Level Set Approach.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17 (2): 158–75. doi:10.1109/34.368173.
- Marella, S., S. Krishnan, H. Liu, and H.S. Udaykumar. 2005. “Sharp Interface Cartesian Grid Method I: An Easily Implemented Technique for 3D Moving Boundary Computations.” *Journal of Computational Physics* 210 (1): 1–31. doi:10.1016/j.jcp.2005.03.031.
- Ma, Zhen, João Manuel R.S. Tavares, Renato Natal Jorge, and T. Mascarenhas. 2010. “A Review of Algorithms for Medical Image Segmentation and Their Applications to the Female Pelvic Cavity.” *Computer Methods in Biomechanics and Biomedical Engineering* 13 (2): 235–46. doi:10.1080/10255840903131878.

- McInerney, Tim, and Demetri Terzopoulos. 1996. “Deformable Models in Medical Image Analysis: A Survey.” *Medical Image Analysis* 1 (2): 91–108. doi:10.1016/S1361-8415(96)80007-7.
- McInerney, T., and D. Terzopoulos. 1995. “Topologically Adaptable Snakes.” In , *Fifth International Conference on Computer Vision, 1995. Proceedings*, 840–45. doi:10.1109/ICCV.1995.466850.
- “Mimics.” 2013. Accessed November 4. <http://biomedical.materialise.com/mimics>.
- Mittal, Rajat, and Gianluca Iaccarino. 2005. “Immersed Boundary Methods.” *Annual Review of Fluid Mechanics* 37 (1): 239–61. doi:10.1146/annurev.fluid.37.061903.175743.
- Mittal, Rajat, and F.M. Najjar. 1999. “Vortex Dynamics in the Sphere Wake.”
- Mittal, R., H. Dong, M. Bozkurtas, F.M. Najjar, A. Vargas, and A. von Loebbecke. 2008. “A Versatile Sharp Interface Immersed Boundary Method for Incompressible Flows with Complex Boundaries.” *Journal of Computational Physics* 227 (10): 4825–52. doi:10.1016/j.jcp.2008.01.028.
- Mousel, John. 2012. “A Massively Parallel Adaptive Sharp Interface Solver with Application to Mechanical Heart Valve Simulations.” *Theses and Dissertations*, January. <http://ir.uiowa.edu/etd/3502>.
- Mukherjee, Dipti Prasad, and Nilanjan Ray. 2012. “Contour Interpolation using Level-set Analysis.” *International Journal of Image and Graphics* 12 (01): 1250004. doi:10.1142/S0219467812500040.
- Pekkan, Kerem, Diane de Zélicourt, Liang Ge, Fotis Sotiropoulos, David Frakes, Mark A. Fogel, and Ajit P. Yoganathan PhD. 2005. “Physics-Driven CFD Modeling of Complex Anatomical Cardiovascular Flows—A TCPC Case Study.” *Annals of Biomedical Engineering* 33 (3): 284–300. doi:10.1007/s10439-005-1731-0.
- Peskin, Charles S. 1972. “Flow Patterns around Heart Valves : A Digital Computer Method for Solving the Equations of Motion /.”
- Pham, Dzung L., Chenyang Xu, and Jerry L. Prince. 2000. “Current Methods in Medical Image Segmentation1.” *Annual Review of Biomedical Engineering* 2 (1): 315–37. doi:10.1146/annurev.bioeng.2.1.315.
- Rinaudo, Antonino, Giuseppe D’Ancona, Jake J. Lee, Gerlando Pilato, Andrea Amaducci, Roberto Baglini, Fabrizio Follis, Michele Pilato, and Salvatore Pasta. 2014. “Predicting Outcome of Aortic Dissection with Patent False Lumen by Computational Flow Analysis.” *Cardiovascular Engineering and Technology* 5 (2): 176–88. doi:10.1007/s13239-014-0182-x.
- Sambasivan, Shiv K., and H. S. Udaykumar. 2011. “A Sharp Interface Method for High-Speed Multi-Material Flows: Strong Shocks and Arbitrary Materialpairs.” *Int. J. Comput. Fluid Dyn.* 25 (3): 139–62. doi:10.1080/10618562.2011.558011.
- Santamarina, A, E Weydahl, J M, Jr Siegel, and J E, Jr Moore. 1998. “Computational Analysis of Flow in a Curved Tube Model of the Coronary Arteries: Effects of Time-Varying Curvature.” *Annals of Biomedical Engineering* 26 (6): 944–54.
- Sethian, James Albert. 1999. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science [...] [...] .* Cambridge [u.a.: Cambridge University Press.

- Shang, Eric K., Derek P. Nathan, Ronald M. Fairman, Joseph E. Bavaria, Robert C. Gorman, Joseph H. Gorman, and Benjamin M. Jackson. 2015. "Use of Computational Fluid Dynamics Studies in Predicting Aneurysmal Degeneration of Acute Type B Aortic Dissections." *Journal of Vascular Surgery* 62 (2): 279–84. doi:10.1016/j.jvs.2015.02.048.
- "ShIRT: Sheffield Image Registration Toolkit." 2016. Accessed February 7. <https://portal.vph-share.eu/resources/dcac98cc-658a-4851-b89b-ea9c5adaf71f/>.
- Shrestha, Liza. 2010. "CFD Study on Effect of Branch Sizes in Human Coronary Artery." *Theses and Dissertations*, January. <http://ir.uiowa.edu/etd/885>.
- Singh, Pankaj K., Alberto Marzo, Bethany Howard, Daniel A. Rufenacht, Philippe Bijlenga, Alejandro F. Frangi, Patricia V. Lawford, Stuart C. Coley, D. Rodney Hose, and Umang J. Patel. 2010. "Effects of Smoking and Hypertension on Wall Shear Stress and Oscillatory Shear Index at the Site of Intracranial Aneurysm Formation." *Clinical Neurology and Neurosurgery* 112 (4): 306–13. doi:10.1016/j.clineuro.2009.12.018.
- "SNAP." 2002. <http://www.midag.unc.edu/software/snap/>.
- Sonka, Milan, Vaclav Hlavac, and Roger Boyle. 2008. *Image Processing, Analysis, and Machine Vision*. Toronto: Thompson Learning.
- Stalling, Detlev, Malte Westerhoff, and Hans-Christian Hege. 2005. "Amira -a Highly Interactive System for Visual Data Analysis." <http://www.zib.de/visual-publications/sources/src-2005/amira-overview.pdf>.
- Steinman, David A., and Charles A. Taylor. 2005. "Flow Imaging and Computing: Large Artery Hemodynamics." *Annals of Biomedical Engineering* 33 (12): 1704–9. doi:10.1007/s10439-005-8772-2.
- Steve, Pieper, Lorenzen Bill, Schroeder Will, and Kikinis Ron. 2006. "The NA-MIC Kit:ITK,VTK,Pipelines, Grids and 3D Slicer as An Open Platform for the Medical Image Computing Community: From Nano to Macro." *IEEE International Symposium on Biomedical Imaging. 2006*.
- Steve, Pieper, and Kikinis Ron. "3dSlicer." *3DSlicer*. <http://www.slicer.org/>.
- Suri, Jasjit S, Kecheng Liu, Sameer Singh, Swamy N Laxminarayan, Xiaolan Zeng, and Laura Reden. 2002. "Shape Recovery Algorithms Using Level Sets in 2-D/3-D Medical Imagery: A State-of-the-Art Review." *IEEE Transactions on Information Technology in Biomedicine: A Publication of the IEEE Engineering in Medicine and Biology Society* 6 (1): 8–28.
- Taylor, C A, and C A Figueroa. 2009. "Patient-Specific Modeling of Cardiovascular Mechanics." *Annual Review of Biomedical Engineering* 11: 109–34. doi:10.1146/annurev.bioeng.10.061807.160521.
- Terzopoulos, Demetri, Andrew Witkin, and Michael Kass. 1988. "Constraints on Deformable Models: Recovering 3D Shape and Nongrid Motion." *Artif. Intell.* 36 (1): 91–123. doi:10.1016/0004-3702(88)90080-X.
- Tse, Kwong Ming, Peixuan Chiu, Heow Pueh Lee, and Pei Ho. 2011. "Investigation of Hemodynamics in the Development of Dissecting Aneurysm within Patient-Specific Dissecting Aneurysmal Aortas Using Computational Fluid Dynamics (CFD) Simulations." *Journal of Biomechanics* 44 (5): 827–36. doi:10.1016/j.jbiomech.2010.12.014.
- Tseng, Yu-Heng, and Joel H. Ferziger. 2003. "A Ghost-Cell Immersed Boundary Method for Flow in Complex Geometry." *Journal of Computational Physics* 192 (2): 593–623. doi:10.1016/j.jcp.2003.07.024.

- Udaykumar, H. S., Sreedevi Krishnan, and Saikrishna V. Marella. 2009. “Adaptively Refined, Parallelised Sharp Interface Cartesian Grid Method for Three-Dimensional Moving Boundary Problems.” *International Journal of Computational Fluid Dynamics* 23 (1): 1–24. doi:10.1080/10618560802660379.
- Udaykumar, H. S., R. Mittal, P. Rampunggoon, and A. Khanna. 2001. “A Sharp Interface Cartesian Grid Method for Simulating Flows with Complex Moving Boundaries.” *J. Comput. Phys* 174: 345–80.
- Van De Vosse, F. N., A. A. Van Steenhoven, A. Segal, and J. D. Janssen. 1989. “A Finite Element Analysis of the Steady Laminar Entrance Flow in a 90° Curved Tube.” *International Journal for Numerical Methods in Fluids* 9 (3): 275–87. doi:10.1002/flid.1650090304.
- Vigmostad, Sarah C., Holavanahalli S. Udaykumar, Jia Lu, and Krishnan B. Chandran. 2010. “Fluid–structure Interaction Methods in Biological Flows with Special Emphasis on Heart Valve Dynamics.” *International Journal for Numerical Methods in Biomedical Engineering* 26 (3-4): 435–70. doi:10.1002/cnm.1340.
- Vincent, L., and P. Soille. 1991. “Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (6): 583–98. doi:10.1109/34.87344.
- Whitaker, R T. 2000. “A Level-Set Approach to Image Blending.” *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society* 9 (11): 1849–61. doi:10.1109/83.877208.
- Wolf, I., Marco Nolden, Thomas Botti, Ingmar Wegner, Max Schobingner, Mark Hastenteufel, Tobias Heimann, Hans-Peter Meinzer, and Marcus Vetter. 2005. “The MITK Approach.” In . *The Insight Journal*. <http://hdl.handle.net/1926/14>.
- Wong, Eric C. 2014. “An Introduction to ASL Labeling Techniques.” *Journal of Magnetic Resonance Imaging* 40 (1): 1–10. doi:10.1002/jmri.24565.
- Xiong, Guanglei, C Alberto Figueroa, Nan Xiao, and Charles A Taylor. 2011. “Simulation of Blood Flow in Deformable Vessels Using Subject-Specific Geometry and Spatially Varying Wall Properties.” *International Journal for Numerical Methods in Biomedical Engineering* 27 (7): 1000–1016. doi:10.1002/cnm.1404.
- Xu, Anping, Lijuan Wang, Sha Feng, and Yunxia Qu. 2010. “Threshold-Based Level Set Method of Image Segmentation.” In *2010 3rd International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, 703–6. doi:10.1109/ICINIS.2010.181.
- Xu, X. Y., N. Sun, D. Liu, and N. B. Wood. 2012. “Patient-Specific Analysis of Blood Flow and Mass Transport in Small and Large Arteries.” In *Patient-Specific Modeling in Tomorrow’s Medicine*, edited by Amit Gefen, 189–215. *Studies in Mechanobiology, Tissue Engineering and Biomaterials* 09. Springer Berlin Heidelberg. http://link.springer.com.proxy.lib.uiowa.edu/chapter/10.1007/8415_2011_95.
- Zalesak, Steven T. 1979. “Fully Multidimensional Flux-Corrected Transport Algorithms for Fluids.” *Journal of Computational Physics*, no. 3: 335–62. doi:10.1016/0021-9991(79)90051-2.