

EXTENSIONS TO THE COMPUTATIONAL HEMODYNAMICS MODELING OF
CEREBRAL ANEURYSMS

by

Fernando Mut
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computational Sciences and Informatics

Committee:



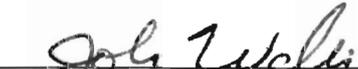
Dr. Juan R. Cebal, Dissertation Director



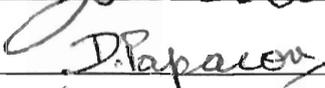
Dr. Rainald Löhner, Committee Member



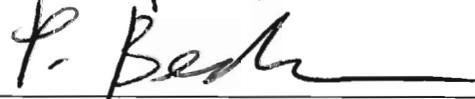
Dr. Timothy Sauer, Committee Member



Dr. John F. Wallin, Committee Member



Dr. Dimitrios A. Papaconstantopoulos,
Department Chair Person



Dr. Peter A. Becker,
Associate Dean for Graduate Programs,
College of Science



Dr. Vikas Chandhoke,
Dean, College of Science

Date: July 31, 2008

Summer Semester 2008
George Mason University
Fairfax, VA

Extensions to the Computational Hemodynamics Modeling of Cerebral Aneurysms

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy at George Mason University

By

Fernando Mut
Master of Science
Instituto Balseiro, 2003
Bachelor of Science
Instituto Balseiro, 2002

Director: Dr. Juan R. Cebal, Professor
Department of Computational and Data Sciences

Summer Semester 2008
George Mason University
Fairfax, VA

Copyright © 2008 by Fernando Mut
All Rights Reserved

Dedication

I dedicate this dissertation to my family and especially to my wife who has supported me during these years.

Acknowledgments

I would like to thank the following people who made this possible.

My advisor, Dr. Juan Cebal for his invaluable support, encouragement, supervision and useful suggestions throughout this research work.

Dr. Rainald Löhner, Director of the Center for CFD at George Mason University for his valuable suggestions throughout this study.

Dr. Romain Aubry for his help with implementing the deflated conjugate gradients method within the incompressible flow solver FEFLO.

Susan Wright and Dr. Giorgio Ascoli from the Krasnow Institute at George Mason University who provided the segmentation of the extended model of the circle of Willis with the attached arterial network.

Dr. Christopher Putman, Director of Interventional Neuroradiology at Inova Fairfax Hospital for his valuable medical insight that has kept us focused on medically relevant issues.

Boston Scientific and Philips Medical Systems for financial support.

Last but not least, my friends here in the US that have made these years very pleasant.

Table of Contents

	Page
List of Figures	vii
Abstract	xi
1 Introduction	1
1.1 Background	1
1.1.1 Cerebral Aneurysms	1
1.1.2 Aneurysms and Hemodynamics	1
1.1.3 Aneurysms and CFD	3
1.2 Computational Hemodynamics Models	4
1.2.1 Anatomical Modeling	4
1.2.2 Blood Flow Modeling	7
1.3 Current limitations	11
1.4 Issues Addressed	15
2 Increasing Model Size	16
2.1 Introduction	16
2.1.1 Motivation	16
2.1.2 Numerical Solutions	17
2.2 Conjugate Gradients	20
2.3 Deflated Conjugate Gradients	25
2.3.1 Previous Work	25
2.3.2 Theoretical Considerations	26
2.3.3 Deflation Subdomains	29
2.4 Benchmarks	31
2.4.1 Pipe Flow	31
2.4.2 Naca 0012	35
2.4.3 von Karman Vortex Street	38
2.5 Hemodynamic Models	41
2.5.1 Internal Carotid Artery Aneurysm	41
2.5.2 Basilar Tip Aneurysm	48
2.5.3 Circle of Willis	54

2.5.4	Cerebral Arteries	61
2.6	Summary	69
3	Virtual Stenting	70
3.1	Introduction	70
3.2	Methodology	71
3.3	Evaluation	75
3.4	Applications	78
3.4.1	Comparison of Stent Designs	78
3.4.2	Model of Aneurysm treated with stent	82
3.5	Summary	86
4	Conclusions and Outlook	87
	Bibliography	89

List of Figures

Figure	Page
1.1 Introduction: 3DRA images and its corresponding computational models. .	6
1.2 Introduction: Typical flow-rate curve used to impose time-dependent boundary conditions.	10
1.3 Introduction: Visualization of a) streamlines, b,c) mean value of wall shear stress $\langle WSS \rangle$, and e,f) oscillatory shear index (OSI)[68].	11
1.4 Introduction: Example of a typical model of the circle of Willis (left) used in current simulations and an extended model of the circle of Willis (right) with the attached arterial network.	12
2.1 DPCG: Smallest eigenmodes of the Laplace operator on a rectangular domain.	29
2.2 Pipe Flow: Surface mesh and deflation domain boundaries for $l = 20$. . .	32
2.3 Pipe Flow: Pressure and Abs(Velocity) for plane $z = 0$	32
2.4 Pipe Flow: Number of iterations required for the PCG solver at each timestep. 33	
2.5 Pipe Flow: Average number of iterations required for the PCG solver as a function of the pipe length.	34
2.6 Pipe Flow: Total CPU time required for 100 timesteps.	34
2.7 NACA 0012: Surface Mesh and Pressure contours.	35
2.8 NACA 0012: Seedpoints selected by hand used to generate 39 deflation subdomains.	36
2.9 NACA 0012: Number of iterations per pseudo timestep required to achieve steady state.	37
2.10 von Karman Vortex Street: Surface mesh of the 3-D model.	38
2.11 von Karman Vortex Street: Surface mesh and Abs(Vel).	39
2.12 von Karman Vortex Street: Deflation subdomains grown from 18 seedpoints.	40
2.13 von Karman Vortex Street: Number of iterations per timestep.	40

2.14 ICA Aneurysm: Deflation boundary subdomains for 50 groups using advancing layers method (left) and 45 groups using the seedpoints alternative (right).	42
2.15 ICA Aneurysm: Pressure (left) and Wall Shear Stress (right) distributions. at $T = 1.4\text{sec}$	43
2.16 ICA Aneurysm: Average number of iterations for the Pressure-Poisson solver at each timestep for all cases (top) and deflation cases only (bottom). . . .	44
2.17 ICA Aneurysm: CPU time at each timestep for all cases (top) and deflation cases only (bottom).	46
2.18 ICA Aneurysm: Total CPU time required for the 200 timesteps. The best speedup obtained was about 3.18 corresponding to 150 groups using the advancing layers technique.	47
2.19 BT Aneurysm: Deflation boundary subdomains for 100 groups using the advancing layers method (left) and 90 groups using the seedpoints alternative (right).	48
2.20 BT Aneurysm: Pressure and Wall Shear Stress distribution at $T = 1.4\text{sec}$	49
2.21 BT Aneurysm: Average number of iterations for the Pressure-Poisson solver at each timestep for all the cases (top) and the deflation cases only (bottom).	50
2.22 BT Aneurysm: CPU time at each timestep corresponding to all the cases (top) and the deflation cases only (bottom).	52
2.23 BT Aneurysm: Timings corresponding to the basilar model of a cerebral aneurysm. The best speedup obtained was about 4.23 corresponding to 90 groups using the seedpoints alternative.	53
2.24 COW: Deflation boundary subdomains for 150 groups using advancing layers method (left) and 131 groups using the seedpoints alternative (right).	56
2.25 COW: Pressure and Wall Shear Stress distributions at $T = 1.4\text{sec}$	57
2.26 COW: Average number of iterations for the Pressure-Poisson solver at each timestep for all cases (top) and deflation cases only (bottom).	58
2.27 COW: CPU time at each timestep for all cases (top) and deflation cases only (bottom).	59
2.28 COW: Timings corresponding to a model of the Circle of Willis. The best speedup obtained was about 4.23 corresponding to 90 groups using the seedpoints alternative.	60

2.29	Cerebral Arteries: Deflation boundary subdomains corresponding to 1000 groups selected by the advancing layers method (top) and 978 groups selected by the seedpoints alternative (bottom).	63
2.30	Cerebral Arteries: Pressure and Wall Shear Stress distribution at time $T = 1.4\text{sec}$	64
2.31	Cerebral Arteries: Huge pressure drop (top) at time $T = 1.4\text{sec}$. due to an high bend in the vessel geometry. This pressure drop also produces a high wall shear stress region (bottom).	65
2.32	Cerebral Arteries: Average number of iterations for the pressure Poisson solver at each timestep for all the cases (top) and the deflation cases only (bottom).	66
2.33	Cerebral Arteries: CPU time at each timestep corresponding to all the cases (top) and the deflation cases only (bottom).	67
2.34	Cerebral Arteries: Timings corresponding to a model of the cerebral arteries. The best speedup obtained was about 3.8 corresponding to 3500 groups using the advancing layers method.	68
3.1	Virtual Stent Deployment: Intermediate stage during the deformation of a cylindrical surface in order to match the given vessel wall. The arrows represents the actual internal and external forces exerted over the cylinder under deformation.	73
3.2	Virtual Stent Deployment: Different stages of the Virtual Stent Deployment methodology.	74
3.3	Phantom Case: Deployed stent (Neuroform) within an idealized model of an stent with its parent vessel made of glass.	76
3.4	Phantom Case: Virtually Deployed stent using the Virtual Stent Deployment methodology.	77
3.5	Comparison of Stent Designs: Original 3DRA image, computational model and three stents deployed into the model.	79
3.6	Comparison of Stent Designs: Streamlines for the various patient specific cases, pre-stented case (top row), with Neuroform stent (second row), left oriented helical stent (third row), and right oriented helical stent (bottom row).	81

3.7	Pipeline: Top row: rotational angiography prior to treatment (left), vascular model (center), deformed cylinder (right). Bottom row: stent design (left), deployed stent (center), zoom up view of the refined mesh near the stent wires (right).	83
3.8	Pipeline: Flow pattern at peak systole before (left) and after (right) stenting.	83
3.9	Pipeline: Conventional (gray background) and virtual (white background) angiogram prior to stenting (left panel) and right after stenting (right panel).	85

Abstract

EXTENSIONS TO THE COMPUTATIONAL HEMODYNAMICS MODELING OF CEREBRAL ANEURYSMS

Fernando Mut, PhD

George Mason University, 2008

Dissertation Director: Dr. Juan R. Cebal

Image-based patient-specific CFD modeling of blood flows is important for better understanding the hemodynamics in cerebral aneurysms and their treatment. Some limitations of current methodologies have been identified, including 1) model size, 2) endovascular device modeling, 3) missing information and 4) extraction of relevant data. This work addressed the first two of these limitations.

First, a Deflated Preconditioned Conjugate Gradients (DPCG) algorithm was developed to accelerate the computation of incompressible flows in the elongated geometries typically encountered in vascular models. This technique has enabled the modeling of the blood flow in complex arterial networks in a timely manner making these models practical for clinical purposes.

Second, a methodology to model stented aneurysms on a patient-specific basis has been developed. This methodology has allowed the computation of the blood flow in cerebral aneurysms after the treatment with stents or other flow diverters.

These two developments have extended the range of applicability of image-based CFD techniques applied to cerebral hemodynamics.

Chapter 1: Introduction

1.1 Background

1.1.1 Cerebral Aneurysms

Intracranial aneurysms are pathological dilatations of cerebral arteries, which are usually located near arterial bifurcations in the circle of Willis [1–3]. The most serious consequence is their rupture and the subsequent intracranial hemorrhage into the subarachnoid space, with an associated high mortality and morbidity rates [4–7]. Intracranial aneurysms are particularly difficult to treat, and often do not produce symptoms before they rupture [8]. Greater availability and improvement of neuroradiological techniques have resulted in more frequent detection of unruptured aneurysms. Preventive surgery is increasingly considered as the therapeutic option. Planning elective surgery requires a better understanding of the process of aneurysm formation, progression, and rupture so that a sound judgment between the risks and benefits of possible therapies can be made.

1.1.2 Aneurysms and Hemodynamics

The genesis, progression and rupture of cerebral aneurysms are not well understood. However, their pathogenesis is believed to be due to the dynamic forces of the blood on a weakened vascular wall. Ultimately, any rupture is the consequence of the inability of the wall to contain the force of the flowing blood. Yet hemodynamic studies have not found evidence of excessive elevations of peak pressure within cerebral aneurysms to explain the wall failure on a purely mechanical basis, and therefore it is believed that a biological alteration of the aneurysmal wall results in its mechanical weakening over time. Evidence of this has been found in histological studies that have shown a decreased number or degeneration

of endothelial cells, degeneration of the internal elastic lamina, and thinning of the medial layer[9,10].

The formation of cerebral aneurysms is believed to be related to an interaction between high flow hemodynamics force and the arterial wall because of several clinical and experimental observations. Cerebral aneurysms are commonly associated with anatomic variations and pathological conditions such as hypoplasia or occlusion of a segment of the circle of Willis[11–15], or high flow arteriovenous malformations[16,17], that cause locally increased flow in the cerebral circulation, and at points of flow bifurcation, a site of flow separation and elevated Wall Shear Stress (WSS). Observations from animal models have shown that elevations of WSS to levels that can be found in these conditions can cause fragmentation of the internal elastic lamina of blood vessels[18] as well as alterations in endothelial phenotype or endothelial damage[9]. Additionally, increased flow and systemic hypertension are required for creating experimental cerebral aneurysms in rats and primates[19–23].

Despite this agreement in the mechanism of aneurysm initiation there is significant controversy regarding the mechanisms responsible for growth and ultimate rupture of a cerebral aneurysm. This controversy can be divided into two main schools of thought: *high* flow effects and *low* flow effects. In each theory, the hemodynamics environment within the aneurysm interacts with the cellular elements of the aneurysmal wall to result in the weakening of the wall. The distinguishing feature between the two schools of thought is in the mechanisms responsible for wall weakening.

The high flow theory focuses on the effects of elevation of WSS. Elevation of maximal WSS can cause endothelial injury and this initiate wall remodeling and potential degeneration[24]. A vascular endothelium malfunction or/and an abnormal shear stress field can cause a lower, non-physiological local arterial tone[25–28]. This would result in a disturbance of the equilibrium between the blood pressure forces and the internal wall stress force, in favor of the first, and a subsequently local dilatation of the arterial wall.

The low flow theory points to low flows within aneurysms as causing localized stagnation of blood flow against the wall. Blood stagnation is known to trigger mechanisms resulting

in the aggregation of red blood cells, as well as accumulation and adhesion of both platelets and leukocytes along the internal surface[29–32]. This process may cause intimal damage, leading to infiltration of white blood cells and fibrin inside the aneurysm wall. The inflammation would lead to localized degeneration of the aneurysm wall that would progressively thin and may result finally into a tearing of the tissue.

1.1.3 Aneurysms and CFD

The flow dynamics of cerebral aneurysms have been studied in numerous experimental models and clinical studies to investigate the role of hemodynamic forces in the initiation, growth, and rupture of cerebral aneurysms[24,33–40]. Although this work has characterized the complexity of intra-aneurysmal hemodynamics in experimental and computational models, the studies have largely focused on idealized aneurysm geometry or surgically created aneurysms in animals. Each of these previous approaches has had significant limitations in connecting the hemodynamics factors studied to clinical events. *In vitro* studies have allowed very detailed measurement of hemodynamic variables[40], but cannot be directly used for understanding the hemodynamic forces in an individual clinical case.

Recent progress in high field phase-contrast MR methods for measuring blood flow dynamics has allowed the visualization and quantification of flow patterns in cerebral arteries *in vivo*[41, 42]. However, these techniques are limited by the image resolution and flow disturbances.

Therefore, realistic image-based computational models can be constructed from anatomical images. This is an attractive alternative because of the ability of computational models to handle any vessel geometry. Image-based CFD has been applied to the study of a variety of vascular diseases[43–48]. In addition, computational models can be used not only to study the current hemodynamics conditions of a given patient (as any imaging modality would do), but also they provide the possibility of asking *what-if* questions. For instance, it is possible to study the alterations of the blood flow patterns of a particular patient induced by surgical procedures such as bypass surgery[49] or endovascular interventions

such as stenting and aneurysm coiling[50]. This opens the possibility of choosing the best therapeutic alternative for a given patient, and also of personalizing and optimizing the treatment for the particular anatomical and hemodynamic structures of each individual. This predictive character of patient-specific image-based computational models cannot be reproduced with any imaging modality.

1.2 Computational Hemodynamics Models

The process of simulation of patient-specific hemodynamics from medical images can be divided into two major stages: a) anatomical modeling and b) blood flow modeling. Anatomical modeling includes two basic steps: 1) image processing and 2) geometrical modeling. Blood flow modeling is also divided in several sub-steps: 1) flow simulation, 2) post processing and 3) visualization. The set of sequential modeling stages is called a computational modeling pipeline or chain.

Several alternative approaches exist for each of the stages of the modeling chain, and different investigators have used different combinations of computational tools to assemble their pipelines. In the center for CFD, our research team have developed the following inhouse software for each step of the pipeline: a) ZMD for anatomical modeling, b) GEN3D for mesh generation, c) FEFLO for the numerical solution of the blood flow and d) ZFEM for the visualization of the results. In what follows, a brief description of this pipeline is provided[51].

1.2.1 Anatomical Modeling

Patient-specific anatomical models can be constructed from a variety of imaging modalities such as 3D rotational angiography (3DRA), computed tomography angiography (CTA) and magnetic resonance angiography (MRA). Rotational angiography is an invasive technique that requires an intra-arterial injection of contrast material and exposition to a low dose of X-rays, but provides the highest resolution and contrast between vascular structures

and the surrounding tissue. CTA is a less invasive technique that requires an intra-venous injection of contrast material and exposition to X-rays. The resolution is less than that of rotational angiography and other non-vascular structures such as bone appear bright in these images, complicating the anatomical modeling process. However, this imaging modality is useful providing information about the peri-aneurysmal environment. MRA does not use any ionizing radiation but suffers from signal loss in regions of decreased or disturbed flow patterns. As in the case of CTA, in MRA images tissues other than blood vessels are also visualized, complicating the segmentation process. The preferred imaging modality for numerical modeling is then the rotational angiography because of its superior quality depicting the vascular structures and simplicity for constructing anatomical models.

After obtaining the clinical data set, the first step in the construction of a patient-specific anatomical model is to filter the anatomical image in order to reduce the noise and increase the contrast between the blood vessels and surrounding tissue. Several image-based techniques could be used to this purpose, including basic ones like sharpening[52] and blurring[53] (isotropic diffusion), and more advanced techniques such as inhomogeneous and anisotropic diffusion methods[54]. Vessel enhancement filters based on the local structure of the image intensity distribution have also been designed[55]. These techniques aim at smoothing out non-vascular structures and at the same time increase the contrast of tubular structures in the images.

The second step in the model construction process is the segmentation of vascular structures. The result of this process is the classification of the image voxels into blood vessels and other tissues. Several methods can be used for this process. The simplest method is thresholding[53], in which all the voxels with an intensity value greater than a given threshold are classified as blood vessels. Another approach is a seeded region growing, which is usually used to segment simple vascular structures [52, 53]. This approach consists in manually selecting a seed voxel within the desired vascular tree and marking all voxels connected to it within a specific intensity range. Since these two techniques are based on the image intensity level, they could fail because in many cases the image intensity distribution

is not homogeneous within the blood vessels and there may be other tissues with similar intensities.

Once the image has been segmented, a geometrical model (surface triangulation) is constructed via iso-surface extraction. This initial surface could be adjusted using the so called deformable models. The basic idea behind this approach is to allow the surface to deform under internal elastic forces between neighboring nodes and external forces derived from the local image intensity gradient[56].

The final step in the model construction is the generation of a 3D volume mesh suitable for computational fluid dynamics simulations. This process starts with the construction of an appropriate surface mesh (triangulation) using the geometrical model as the starting point. Once the surface mesh is obtained, the space within the anatomical model is filled with tetrahedral elements using the advancing front method[57].

Figure (1.1) shows some examples of anatomical models constructed from 3DRA images using this technique.

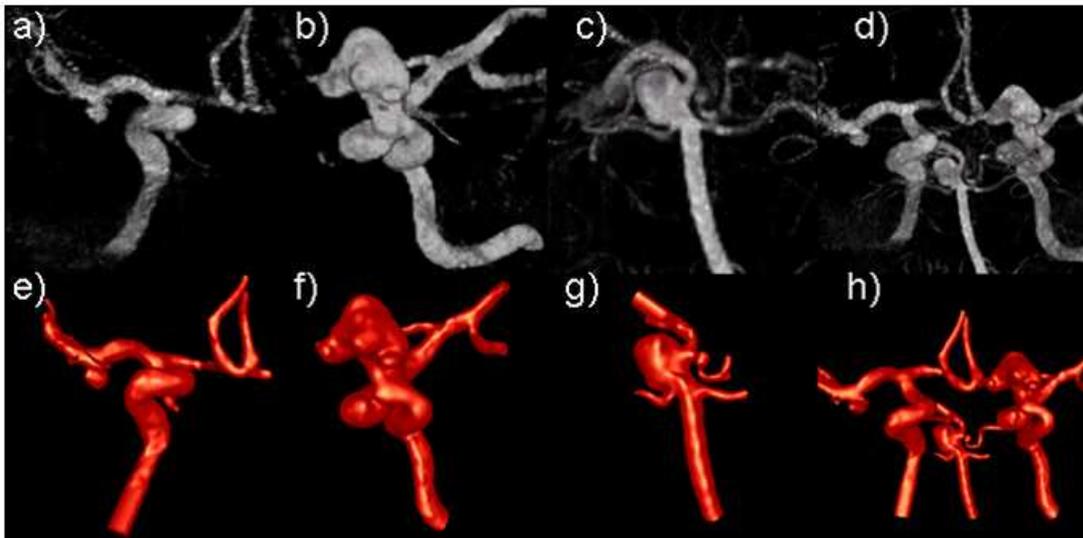


Figure 1.1: Introduction: 3DRA images and its corresponding computational models.

1.2.2 Blood Flow Modeling

Governing Equations

Blood flow is mathematically modeled by the unsteady Navier-Stokes equations for an incompressible fluid[57–59]:

$$\nabla \cdot \mathbf{v} = 0 \quad (1.1)$$

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v} \cdot \nabla)\mathbf{v} + \nabla p = \nabla \cdot \boldsymbol{\tau} + \mathbf{f} \quad (1.2)$$

where ρ denotes the (constant) density, p the pressure, \mathbf{v} the velocity vector and $\boldsymbol{\tau}$ the deviatoric stress tensor. Although the stress/strain-rate relationship is a tensor relation, it is usually expressed as an algebraic equation of the form:

$$\boldsymbol{\tau} = \mu \dot{\boldsymbol{\gamma}} \quad (1.3)$$

where μ is the viscosity, and the strain-rate is defined as the second invariant of the strain-rate tensor, which for incompressible fluids is:

$$\dot{\boldsymbol{\gamma}} = 2\sqrt{\boldsymbol{\epsilon}_{ij}\boldsymbol{\epsilon}_{ij}} \quad ; \quad \boldsymbol{\epsilon}_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \quad (1.4)$$

In order to close the system of equations, a constitutive law must be provided to compute the local viscosity of the fluid. The simplest rheological model is a Newtonian fluid, which assumes a constant viscosity: $\mu = \mu_0$. Typical values used for blood are $\rho = 1.105 \text{ g/cm}^3$ and $\mu = 0.04 \text{ Poise}$.

However, blood can be thought of as a suspension of particles (red blood cells) in an aqueous medium (plasma). Thus, it is neither homogeneous nor Newtonian. The rheological properties of blood are mainly dependent on the maturity, or the volume fraction of red blood cells in the blood. One of the most commonly used non-Newtonian fluid models for

blood is the model of Casson[60], which assumes a stress/strain-rate relation of the form

$$\sqrt{\tau} = \sqrt{\tau_0} + \sqrt{\mu_0} \sqrt{\dot{\gamma}} \quad (1.5)$$

where τ_0 is the yield stress and μ_0 is the Newtonian viscosity. The existence of a yield stress implies that blood requires a finite stress before it begins to flow, a fact that has been observed experimentally. The apparent viscosity of the Casson model can be written as

$$\mu = \left(\sqrt{\frac{\tau_0}{\dot{\gamma}}} + \sqrt{\mu_0} \right)^2 \quad (1.6)$$

Since this expression diverges as the strain-rate becomes zero, it is typically modified in the following way:

$$\mu = \left[\sqrt{\tau_0 \left(\frac{1 - e^{-m\dot{\gamma}}}{\dot{\gamma}} \right)} + \sqrt{\mu_0} \right]^2 \quad (1.7)$$

where the parameter m controls the maximum viscosity obtained when $\dot{\gamma}$ tends to zero. Typical values used for blood are $\tau_0 = 0.04$ dyne/cm², $\mu_0 = 0.04$ dyne/cm and $m = 100$.

Boundary Conditions

In order to obtain unique solutions to the Navier-Stoke equations, proper boundary conditions need to be specified. These boundary conditions attempt to provide the computational model with information about the surroundings of the anatomical model. For the inflows/outflows of the model, typically the pressure and/or the velocity profile is prescribed. For the vessel wall the simplest model consists on a rigid wall. Another possibility is to consider the vessel wall as an elastic membrane, for which a solver for the elasticity equations has to be coupled with the flow solver. The proper specification of the boundary conditions for a given model depends on the amount of information available in each case. Since the physiological parameters can not be obtained for all patients, some assumptions

have to be made.

Physiologic velocity profiles for the boundaries are derived from phase-contrast magnetic resonance (MR) images of the main branches of the circle of Willis [47,61]. Time-dependent flow rates are obtained by integration of the measured velocity profile over the vessel cross section. The curve is then decomposed into Fourier modes:

$$Q(t) = \sum_{n=0}^N Q_n e^{inwt} \quad (1.8)$$

where N is the number of modes and w the angular frequency obtained from the period of the cardiac cycle. The imposed velocity profile is then computed from the Womersley solution [62,63]. This profile is mapped to the boundary using an algorithm that maps the surface mesh of the boundary to a circle [61]. Figure 1.2 shows a typical flow rate curve used in the simulations.

Since flow rates are not always available for all the branches included in the models, traction-free boundary conditions are applied in all the remaining boundaries. It is known that the flow divisions are determined by the impedance of the distal arterial tree. Imposing traction-free boundary conditions for all the outflows boundaries assumes that the corresponding distal resistances are similar, and therefore the flow division is determined by the geometry of the anatomical model. This is not an unrealistic assumption when the outflow vessels are of similar calibers.

Vessel wall compliance is an important effect that may alter the local hemodynamics. Fluid-Solid interaction models have been applied to the study of these effects on the flow patterns [64]. However, the main problem remains the proper characterization of the arterial wall, i.e., the values of local material properties such as the modulus of elasticity, wall thickness, etc. In addition, coupled fluid-solid models require the specification of proper pressure boundary conditions, i.e. the pressure waveform that drives the motion of the arterial wall, which is difficult to measure noninvasively [65]. In addition, a sensitivity

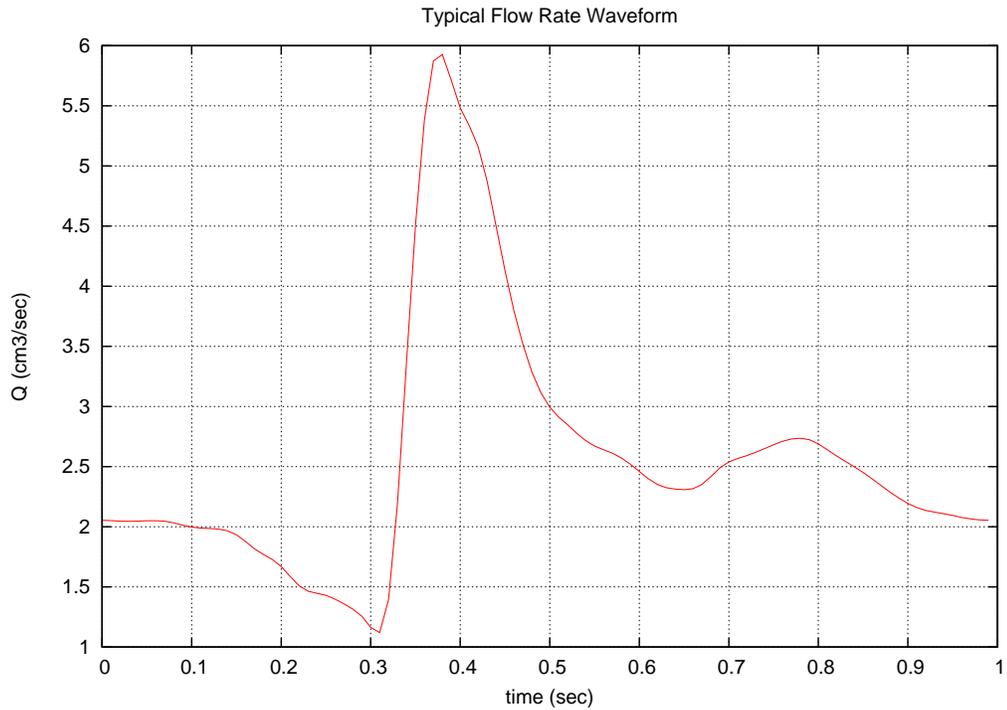


Figure 1.2: Introduction: Typical flow-rate curve used to impose time-dependent boundary conditions.

analysis carried out using dynamic imaging of the vascular wall showed that the main hemodynamics characteristics computed with rigid and deforming wall models are in good agreement[66, 67] For these reasons, the vessel walls are usually assumed rigid. No-slip boundary conditions are applied at the vessel walls, i.e., the fluid velocity is equal to the velocity of the wall, which under the assumption of rigid vessel walls becomes $\mathbf{v} = 0$.

Post-processing and Visualization

The last step in the modeling pipeline is the analysis of the blood flow simulations and extraction of relevant data. This analysis can be divided into two sub-steps: 1) post-processing and 2) visualization. Post-processing involves further calculations using the blood flow solution in order to obtain derived data sets that are helpful in the characterization and analysis of the flow patterns. These derived data sets include the wall shear stress (WSS), streamlines, transport of scalars (e.g. virtual angiograms), etc. Statistical computations

are also carried out to obtain information like mean values, peaks, etc. All the computed quantities are then visualized using advanced 3D visualization techniques and graphics libraries (OpenGL). Figure (1.3) shows some examples of visualizations of such quantities.

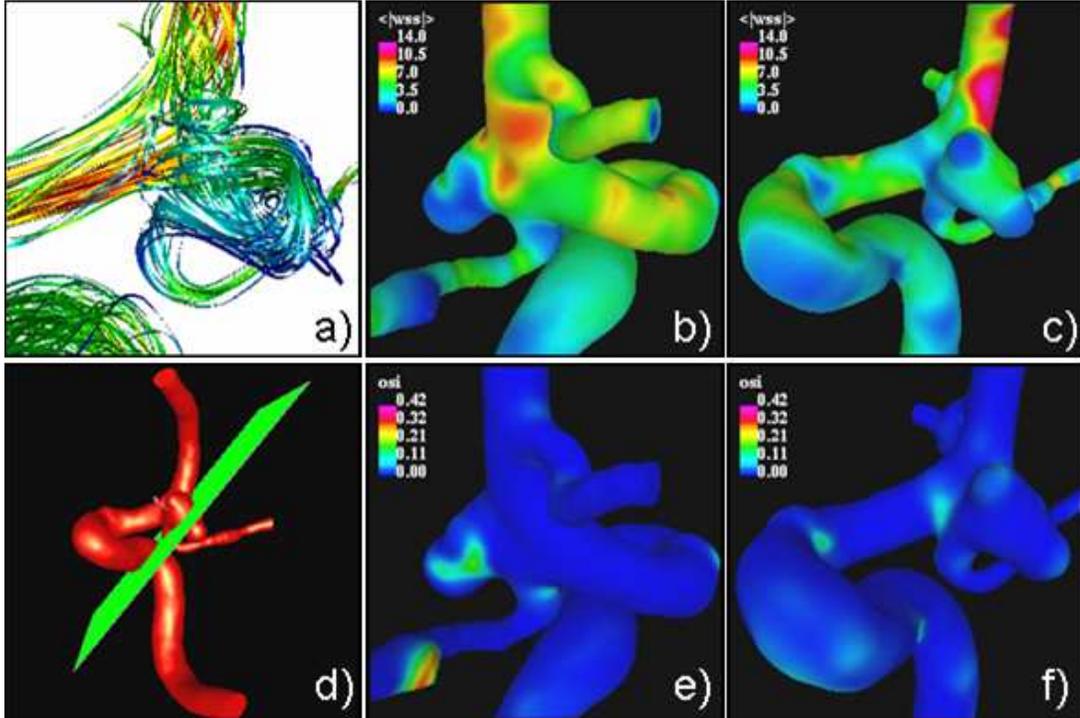


Figure 1.3: Introduction: Visualization of a) streamlines, b,c) mean value of wall shear stress $\langle |WSS| \rangle$, and e,f) oscillatory shear index (OSI)[68].

1.3 Current limitations

The computational modeling pipeline described in the previous sections have proven to be a powerful tool in helping to understand the role of the hemodynamics forces in the initiation, growth, and rupture of cerebral aneurysms[44,48,49]. However, there are several issues that could be addressed in order to improve the actual state of the computational modeling of cerebral aneurysms.

In what follows a brief description of the issues that we believe are the key factors towards the next generation of computational hemodynamics models of cerebral aneurysms

is presented.

1. Increasing model size:

With the constant improvement of medical imaging equipment physicians are able to produce higher resolution imaging studies. This results in highly detailed anatomical models which implies a bigger number of elements in order to achieve an accurate spatial discretization of the domain. This positively enhances the quality of the blood flow simulation for a particular model. However the convergence rate of the flow solver is severely deteriorated.

On the other hand, it is also desired to enhance the current models by considering larger vascular structures upstream and downstream (proximal and distal) to the aneurysm. However, this implies a significant increase of the model size, which is followed by slower convergence rates of the flow solver. Figure 1.4 shows an example of a typical model for the circle of Willis (left) and an extended version which includes the downstream arterial network (right).

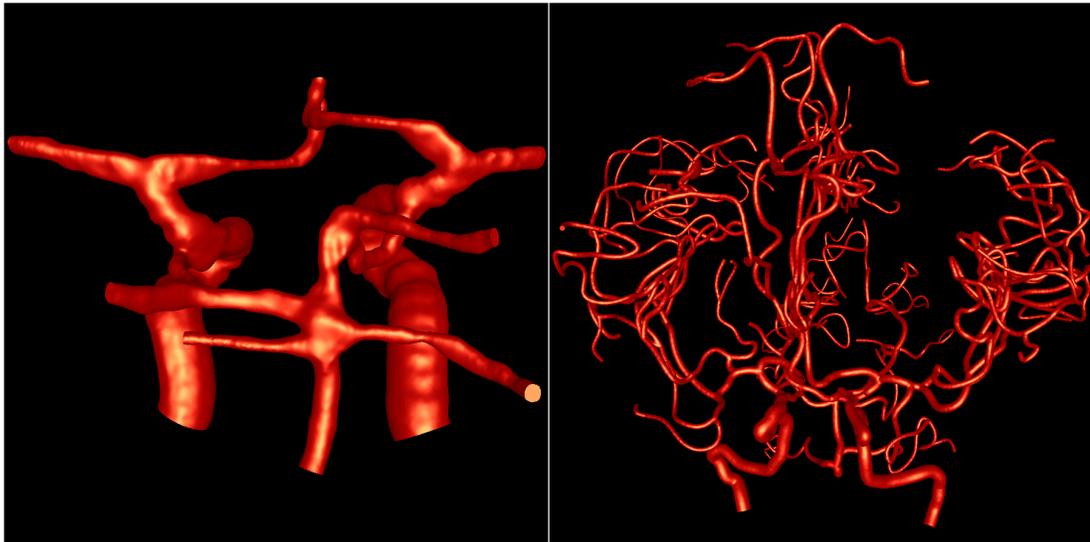


Figure 1.4: Introduction: Example of a typical model of the circle of Willis (left) used in current simulations and an extended model of the circle of Willis (right) with the attached arterial network.

The ability to produce fast and accurate results in patient-specific models is desired since it provides the possibility of modeling many cases in a timely manner and thus obtain reliable statistical results. This also will allow physicians to model new patients in a daily basis in order to quickly assess which are the best treatment options available.

2. Deployment of vascular devices:

Currently there is a increased interest in using endovascular devices for flow diversion (a.k.a flow diverters) such as stents, for treating cerebral aneurysms. These devices are intended to divert the blood flow away from the aneurysms in order to prevent their future rupture. Several studies have proven that the final position of the stent, as well as its geometrical design, plays a major role in the way the blood flow is diverted. For this reason, a computational model to study the changes in the blood flow patterns after stenting is a required tool. This will allow the design of better flow diverters and the selection of the most appropriate one for a given aneurysm.

One of the most important steps in patient-specific virtual stenting modeling is the stent deployment within a given anatomical model. Also the ability to easily interchange different stent designs within the same model is desired in order to perform fast comparative characterizations.

3. Extraction of relevant information:

The objective of the computational hemodynamics modeling of cerebral aneurysms is to identify matches or links between hemodynamic variables (or characteristics) of a given aneurysm and clinical events. This requires the identification of the relevant hemodynamics quantities that play a major role in the genesis and progression of cerebral aneurysms. In order to accomplish this, a complete characterization of the hemodynamics of cerebral aneurysms has to be performed.

The solution of the governing equations for the blood flow results in a detailed description of the velocity and pressure fields within the anatomical model as a function of

time. The solution fields are further processed (post-processing step) in order to obtain derived quantities that are believed to have relevance in the processes that drive the genesis of the aneurysms. Example of these quantities are the wall shear stress (WSS), and the impingement region, i.e., the region in which the jet flow impacts the wall of the aneurysm.

Currently the researcher performs a visual inspection of the computed quantities in order to obtain a *qualitative* characterization of the hemodynamics for a given aneurysm (see figure 1.3). This characterization classifies the hemodynamics into four types: 1) simple stable , 2) simple unstable, 3) complex stable and 4) complex unstable. Although this procedure has been successful in identifying some trends in the progression of cerebral aneurysms[48], a better and *quantitative* characterization is required. One of the main benefits of such a tool would be to answer questions like “how different are these two aneurysms?”, “what are the effects of a given stent design or treatment option?”, and “how close is this aneurysms to a particular class?”.

4. Missing information:

A realistic modeling of the hemodynamics of cerebral aneurysms involves the knowledge of several patient-specific physiological parameters. This information is then included in the flow solver as boundary conditions, material properties, etc. Among the most important that can be identified are the blood flow inflow rate and the outflow boundary conditions.

Since many of these physiological parameters can not be measured for all patients, some assumptions have to be made. Currently for the inflow rate, measurements done on normal subjects are used after a proper scaling based on the actual patient’s vessel area[69] For the outflow boundary conditions, a simple constant pressure is assumed. However the vascular network (see figure 1.4) that is attached at the outflow of the model behaves more like a resistant device. This suggest that more realistic simulations will require taking in consideration the study of resistance-based boundary

conditions. Although recent studies[70] have shown how to deal with this type of boundary conditions, a deeper analysis of the problem, as well as the development of new techniques is needed.

1.4 Issues Addressed

Within the context of this work we have made substantially progress to solve two of the four current limitations presented in the previous section, namely, 1) the increasing model size problem, and 2) the deployment of vascular devices (virtual stent deployment).

For the increasing model size problem we have developed a *Deflated Preconditioned Conjugate Gradients* (DPCG) technique[71] which has successfully accelerated the convergence rate of the pressure Poisson equation within the incompressible flow solver. Several tests have shown an speedup between 3 and 5 of the total computational time when compared with the traditional method (standard Preconditioned Conjugate Gradients). This technique has enabled the possibility of computing a pulsatile solution of the hemodynamics for a given cerebral aneurysm in a few hours and a steady calculation in a few minutes. At the same time it has provided the ability of modeling much more complex structures such as a complete vascular model of the circle of Willis together with the arterial network of the brain.

For the deployment of vascular devices we have developed a pioneer technique for the virtual deployment of stents within patient-specific anatomical models of cerebral aneurysms. This technique has allowed to realistically study the changes in the blood flow patterns before and after stenting. One of the main advantages of this new methodology is the ability to easily interchange different stent designs in order to do fast comparative characterizations. Coupled with the DCG technique, it will allow the physicians to better evaluate a treatment plan for a given patient in a daily basis.

In the following chapters details about these new techniques are provided. The results of several evaluation studies that have been carried out are presented, as well as several examples of these methodologies applied to patient-specific models.

Chapter 2: Increasing Model Size

2.1 Introduction

2.1.1 Motivation

The computational hemodynamic modeling of cerebral aneurysms involves the solution of the unsteady Navier-Stokes equations for an incompressible fluid:

$$\nabla \cdot \mathbf{v} = 0 \tag{2.1}$$

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v} \cdot \nabla)\mathbf{v} + \nabla p = \nabla \mu \cdot \nabla \mathbf{v} \tag{2.2}$$

where ρ denotes the (constant) density, p the pressure, \mathbf{v} the velocity vector and μ the viscosity.

What sets incompressible flow solvers apart from compressible flow solvers is the fact that the pressure is not obtained from an equation of state but from the divergence constrain. The so-called pressure Poisson equation can be obtained by taking the divergence of (2.2) and using (2.1):

$$\nabla^2 p = -\nabla \cdot \mathbf{v} \nabla \mathbf{v} \tag{2.3}$$

This implies that the pressure field establishes itself instantaneously (reflecting the infinite speed of sound assumption of incompressible fluids) and must, therefore, be integrated implicitly in time[57]. The fact that the pressure field establishes instantaneously is reflected by the elliptic character of the Pressure Poisson equation. The main implication is that changes on the prescribed values need to be propagated through the entire domain in order to obtain a solution of the Poisson equation. Since the solution of this equation is

usually carried out using an iterative scheme applied to a finite element approximation of the Laplace operator, the number of iterations required to transmit information is bounded from below by the number of elements in the longest direction (graph depth). This fact imposes a severe limitation, particularly in the context of modeling cerebral aneurysms, since typical vascular structures resembles elongated or tubular domains with a corresponding large graph depth.

The motivation of this work was to develop techniques that can accelerate the transmission of information along large tubular-shaped domains. The first attempt in this direction considered the approximation of the solution of the 3D problem by the solution of the associated 1D problem, provided that in these two settings the solutions are similar (in cylindrical domains). The problem with this approach is that the resulting preconditioned matrix was singular. The a deflated preconditioned conjugate gradients algorithm was explored. This method provided the proper framework needed to communicate information along the longest direction by means of the subdomain deflation technique. This method turned out to be equivalent to solving the previously proposed preconditioning technique based on the 1D model but avoiding the singularity in the preconditioning matrix. In what follows a description of the numerical methods used for the discretization of the Navier-Stokes equations is presented.

2.1.2 Numerical Solutions

Several numerical schemes have been used to solve the incompressible Navier-Stokes equations [72–74]. The hyperbolic character of the advection operator and the elliptic character of the pressure-Poisson equation have led to a number of so-called projection schemes. The idea is to predict first a velocity field without taking into account the divergence constraint. The divergence constraint is then enforced by solving the pressure-Poisson equation. The

velocity increment can therefore be separated into an advective and pressure increment:

$$\begin{aligned}\mathbf{v}^{n+1} &= (\mathbf{v}^n + \Delta\mathbf{v}^a) + \Delta\mathbf{v}^p \\ &= \tilde{\mathbf{v}} + \Delta\mathbf{v}^p\end{aligned}\tag{2.4}$$

For an explicit integration of the advective terms, one complete timestep is given by:

(a) *advective-diffusive* prediction: $\mathbf{v}^n \rightarrow \tilde{\mathbf{v}}$

$$\left[\frac{1}{\Delta t} - \nabla\mu\nabla \right] (\tilde{\mathbf{v}} - \mathbf{v}^n) + \mathbf{v}^n \cdot \nabla\mathbf{v}^n + \nabla p^n = \nabla\mu\nabla\mathbf{v}^n\tag{2.5}$$

(b) *pressure* correction: $p^n \rightarrow p^{n+1}$

$$\begin{aligned}\nabla \cdot \mathbf{v}^{n+1} &= 0 \\ \frac{\mathbf{v}^{n+1} - \tilde{\mathbf{v}}}{\Delta t} + \nabla(p^{n+1} - p^n) &= 0\end{aligned}$$

which, by taking the divergence, results in

$$\nabla^2(p^{n+1} - p^n) = \frac{\nabla \cdot \tilde{\mathbf{v}}}{\Delta t}\tag{2.6}$$

(c) *velocity* correction: $\tilde{\mathbf{v}} \rightarrow \mathbf{v}^{n+1}$

$$\mathbf{v}^{n+1} = \tilde{\mathbf{v}} - \Delta t \nabla(p^{n+1} - p^n)\tag{2.7}$$

At a steady state, the residuals of the pressure correction vanish, implying that the result does not depend on the timestep Δt .

The most expensive step within this scheme is the solution of the Poisson equation for the pressure increment, given by equation (2.6). The discretization of this elliptic partial

differential equation via the finite element method leads to a very large, but symmetric linear system of algebraic equations of the form:

$$Ax = b \tag{2.8}$$

Although the system (2.8) is very large (the number of unknowns equals the number of grid points in the computational domain), the number of non-zero entries in the matrix A is comparatively small. This is due to the local support of the interpolant functions used in the finite element method. For simplices (e.g. triangles in 2-D, tetrahedrals in 3-D) there is a one-to-one correspondence between the nonzero coefficients in the matrix and the edges of the computational grid. A grid point in a typical triangular mesh is connected in average to 6 other grid points, i.e., in average there are 6 non-zero entries per row in the matrix. The ratio between the total number of entries in the matrix and the number of non-zero coefficients is known as the sparsity level of the matrix. When the sparsity level is sufficiently large, iterative solvers become the preferred method in solving such systems. For the particular case in which the system is also symmetric positive definite (SPD), one of the best known iterative solvers is the preconditioned conjugate gradients (PCG).

Many attempts have been made to mitigate the impact of the pressure Poisson equation on the overall cost of a simulation. Options that have proven useful include:

- Improved prediction of the starting value for the iterative solver [75, 76].
- Linelet preconditioners for highly stretched grids (e.g. boundary layers) [77, 78].
- Multistage or implicit treatment of the advective terms [79, 80].

Several attempts have also been made to use multigrid solvers [81–86]. However, for unstructured grids the expected gains have proven elusive to date. Moreover, cases with moving and or adapting meshes place further burdens on multigrid solvers vis a vis conjugate gradient solvers.

In what follows a detailed description of the classical Conjugate Gradients (CG) method together with a convergence analysis is presented. After that a full description of the Deflated Conjugate Gradients (DCG) method is presented. The principal aim of the DCG method is to remove from the search space those components of the error that may contribute to a slow convergence rate of the classical CG method. This is carried out by a subspace deflation technique which attempts to approximate the smallest eigenmodes of the system by a coarse discretization of the domain. This technique has proven to be very useful especially when dealing with high-aspect ratio domains. Finally, the results of several experiments that have been carried out to assess the performance of the DCG method are presented.

2.2 Conjugate Gradients

The Conjugate Gradients (CG) method was introduced by Hestenes and Stiefel[87] in 1952 for solving sparse symmetric positive definite (SPD) systems of linear equations. The CG method is a special case of a Conjugate Directions (CD) method, in which the search directions are constructed from the residual vectors in a progressive manner. This choice minimizes the cost in both memory and floating-point operations per iteration of the classical CD method, making the CG algorithm the preferred choice for solving very large but sparse SPD systems.

A classical CD method starts with a full-set of *conjugate* vectors:

$$\{d_k\}_{k=1}^n \in \mathbb{R}^n \quad \text{where} \quad (d_i, Ad_j) = 0 ; \quad i \neq j \quad (2.9)$$

i.e., vectors that are orthogonal with respect to the inner-product defined by A : $(\cdot, \cdot)_A := (\cdot, A(\cdot))$ (well defined since A is SPD). This set of *conjugate* vectors can be obtained from any set of linear independent (L.I.) vectors by the *Conjugate* Gram-Schmidt process (CGS) (see algorithm 2.2.1).

Algorithm 2.2.1 *Conjugate* Gram-Schmidt process

```
1: Given  $\{u_k\}_{k=1}^n$  L.I., set  $d_1 := u_1$ 
2: for  $i = 2$  to  $n$  do
3:    $d_i := u_i$ 
4:   for  $j = 1$  to  $i - 1$  do
5:      $\beta_j = (u_i, Ad_j)/(d_j, Ad_j)$ 
6:      $d_i := d_i - \beta_j d_j$ 
7:   end for
8: end for
```

In other words, for each new vector u_i , the *Conjugate* Gram-Schmidt process subtracts from u_i its A -orthogonal projection with respect to all previously generated vectors $\{d_j\}_{j=1}^{i-1}$. This process will not break (in exact arithmetic) unless the $\{u_k\}_{k=1}^n$ are not L.I. The modified *conjugate* Gram-Schmidt process (MCGS) can be obtained by replacing u_i by d_i in step 5 of algorithm 2.2.1, which in presence of round-off error is much more reliable.

Given an initial guess x_1 , the next step in the CD process is to write the initial error $e_1 = x_* - x_1$ as a linear combination of the conjugate basis vectors:

$$e_1 = \alpha_1 d_1 + \dots + \alpha_n d_n \quad (2.10)$$

The coefficients α_k can be obtained by first taking the A -inner product between equation (2.10) and an arbitrary basis vector d_k :

$$\begin{aligned} Ae_1 &= \alpha_1 Ad_1 + \dots + \alpha_n Ad_n \\ (d_k, Ae_1) &= \alpha_1 (d_k, Ad_1) + \dots + \alpha_n (d_k, Ad_n) \end{aligned} \quad (2.11)$$

After that, applying the conjugacy conditions of the basis vectors d_k yields:

$$\alpha_k = \frac{(d_k, r_1)}{(d_k, Ad_k)} \quad (2.12)$$

where $r_1 = Ae_1$ is the initial residual vector.

This process can be written in a progressive manner, i.e., at each iteration of CD one

component of the initial error is eliminated, allowing to stop the process as soon as a convergence criteria is reached. This is carried out by first defining the following recurrence relations:

$$x_{k+1} = x_k + \alpha_k d_k \quad (2.13)$$

$$e_{k+1} = e_k - \alpha_k d_k \quad (2.14)$$

$$r_{k+1} = r_k - \alpha_k A d_k \quad (2.15)$$

where equation (2.14) is obtained by subtracting the exact solution x_* from equation (2.13), and equation (2.15) is obtained by simply multiplying equation (2.14) from the left by A . Note also that terms of the form $\alpha_j(d_k, Ad_j)$, $1 \leq j < k$ can be added to the left hand side of equation (2.11), since they are all zero by the conjugacy conditions, which by means of (2.14) leads to:

$$\alpha_k = \frac{(d_k, r_k)}{(d_k, Ad_k)} \quad (2.16)$$

where now the coefficients α_k depend only on values at step k .

A Conjugate Directions algorithm will read as follows:

Algorithm 2.2.2 Conjugate Directions (CD)

- 1: Given A , b , $\{u_k\}_{k=1}^n$ L.I. and an initial guess x_1
 - 2: Compute $r_1 := b - Ax_1$
 - 3: Set $d_1 := u_1$
 - 4: **do until** convergence
 - 5: $\alpha_k := (d_k, r_k)/(d_k, Ad_k)$
 - 6: $x_{k+1} := x_k + \alpha_k d_k$
 - 7: $r_{k+1} := r_k - \alpha_k Ad_k$
 - 8: $d_{k+1} := u_{k+1}$
 - 9: **for** $j = 1$ to $k - 1$ **do**
 - 10: $\beta_j = (d_{k+1}, Ad_j)/(d_j, Ad_j)$
 - 11: $d_{k+1} := d_{k+1} - \beta_j d_j$
 - 12: **end for**
 - 13: **end do**
-

After k steps of CD, the error is expressed by:

$$e_{k+1} = \alpha_{k+1}d_{k+1} + \dots + \alpha_n d_n \quad (2.17)$$

Taking the A -inner product of 2.17 with a basis vector d_j , and applying the conjugacy conditions gives:

$$(d_j, Ae_{k+1}) = (d_j, r_{k+1}) = 0 \quad 1 \leq j \leq k \quad (2.18)$$

i.e., e_{k+1} is A -orthogonal to $\mathcal{D}_k = \text{span}\{d_j\}_{j=1}^k$, or equivalently, r_{k+1} is orthogonal to \mathcal{D}_k .

This implies the two following optimality properties:

$$\|r_{k+1}\| = \|b - Ax_{k+1}\| = \min_{x \in x_1 + \mathcal{D}_k} \|b - Ax\| \quad (2.19)$$

$$\|e_{k+1}\|_A = \|x_* - x_{k+1}\|_A = \min_{x \in x_1 + \mathcal{D}_k} \|x_* - x\|_A \quad (2.20)$$

i.e., after k steps a CD method finds the best approximate solution in the *affine* space $x_1 + \mathcal{D}_k$. In other words, a CD process produces at each step the best approximation, in terms of the error and residual, within the provided search space \mathcal{D}_k . Note also that the exact solution is obtained in *at most* n steps. However, the main drawback of a conjugate directions method is that all the previous search directions have to be saved in order to perform the A -orthogonalization process (steps 9-12 of algorithm 2.2.2). Fortunately, this can be avoided by simply replacing the original $\{u_k\}_{k=1}^n$ basis vectors with the residual vectors $\{r_k\}_{k=1}^n$, which is known as the Conjugate *Gradients* (CG) method. In this case, the following properties can be easily verified:

- since $(d_j, r_{k+1}) = 0$ for $1 \leq j \leq k \Rightarrow$ L.I. of r_{k+1} with respect to \mathcal{D}_k is guaranteed, unless $r_{k+1} \equiv 0$, in which case we have the exact solution.
- since $\text{span}\{d_j\}_{j=1}^k = \text{span}\{r_j\}_{j=1}^k \Rightarrow (r_j, r_{k+1}) = 0$ for $1 \leq j \leq k$

Replacing the recurrence relation (2.15) for r_j in the last identity yields:

$$\alpha_j(d_j, Ar_{k+1}) = (r_j, r_{k+1}) - (r_{j+1}, r_{k+1}) = 0 \quad 1 \leq j \leq k-1 \quad (2.21)$$

i.e., the new residual vector r_{k+1} is *already* A -orthogonal with all the previous search directions, except for the last one. This reduces the Conjugate Gram Schmidt orthogonalization process to just one projection, i.e., only against the last search direction. Further simplifications can be achieved by using the recurrence relations, which results in the following expressions for the coefficients:

$$\alpha_k = \frac{(r_k, r_k)}{(d_k, Ad_k)} \quad \beta_k = \frac{(r_{k+1}, r_{k+1})}{(r_k, r_k)} \quad (2.22)$$

The Conjugate Gradients algorithm reads as follows:

Algorithm 2.2.3 Conjugate Gradients (CG)

- 1: Given A , b and an initial guess x_1
 - 2: Compute $r_1 := b - Ax_1$
 - 3: Set $d_1 := r_1$
 - 4: **do until** convergence
 - 5: $\alpha_k := (r_k, r_k)/(d_k, Ad_k)$
 - 6: $x_{k+1} := x_k + \alpha_k d_k$
 - 7: $r_{k+1} := r_k - \alpha_k Ad_k$
 - 8: $\beta_k := (r_{k+1}, r_{k+1})/(r_k, r_k)$
 - 9: $d_{k+1} := r_{k+1} + \beta_k d_k$
 - 10: **end do**
-

In terms of storage, the CG algorithm requires only four vectors (x , d , Ad and r) to be saved. Preconditioning can be easily included in the CG method by simply replacing the usual Euclidean inner product with the M -inner product, where M is an SPD preconditioner matrix.

The convergence behavior of the CG algorithm can be analyzed by exploiting the optimality properties presented above. Let x_k be the approximate solution obtained at the k -th step of the CG algorithm, and x_* be the exact solution. The classical a priori bound

for the error is[88]:

$$\|x_* - x_k\|_A \leq 2 \left[\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right]^k \|x_* - x_1\|_A \quad (2.23)$$

where $\kappa = \lambda_{max}/\lambda_{min}$ is the spectral condition number of the matrix A . Equation 2.23 shows that the convergence rate of the CG algorithm is bounded by the ratio between the biggest and smallest eigenvalues of the underlying matrix. However, as described in Sluis et al. [89], the convergence speeds up as soon as the lowest eigenvalues are 'discovered' by the CG process, giving rise to a condition number based on the active, i.e., the non-discovered eigenvalues. Therefore, if some knowledge of the eigenmodes associated to the smallest eigenvalues is at hand, removing them from the spectrum of A would improve the convergence-rate of the CG process. This is what the Deflated Conjugate Gradients (DCG) tries to achieve.

2.3 Deflated Conjugate Gradients

2.3.1 Previous Work

The first paper to consider a deflation method for the CG solver is perhaps due to Nicolaides[90]. The main idea was to remove certain components of the initial residual that may impede convergence. It also introduced the idea of subdomain deflation, where the lowest eigenmodes are approximated by a coarse discretization of the domain. However, no numerical experiments were reported. The subdomain deflation approach was successfully applied to the bending of a cantilever beam and to the stationary Stokes problem, as reported in Mansfield[91]. The deflation technique was also applied to precondition a Schur complement matrix[92] for second order operators arising from the partition used for parallel computing on each processor. Deflation was also used for an augmented conjugate gradients[93] where the Krylov subspace generated by a previous system is recycled for further solves in subsequent systems. An approximation of the eigenvectors is used to augment the space of subsequent systems to deflate the lowest eigenmodes[94]. In this case the eigenvectors are

computed explicitly and not approximated through a coarse discretization of the domain.

2.3.2 Theoretical Considerations

The Deflated Conjugate Gradients (DCG) technique starts by selecting a subspace $\mathbb{W} \in \mathbb{R}^n$ which is called the *deflation subspace*. Recall that the standard CG algorithm generates a sequence of search directions that are A -orthogonal to all the previous search directions. This motivates the next step of the deflation technique that consists in writing the solution space as a direct sum of \mathbb{W} and its A -orthogonal complement:

$$\mathbb{R}^n = \mathbb{W} \oplus \mathbb{W}^{\perp A} \quad (2.24)$$

Let $\{u_k\}_{k=1}^m$ be a basis of \mathbb{W} . Applying the CGS algorithm (2.2.1) to these vectors produces a conjugate basis $\{d_k\}_{k=1}^m$ of \mathbb{W} . Note that this conjugate basis can be completed by simply adding $(n - m)$ more L.I. vectors to the CGS process. The initial error $e_0 = x_* - x_0$ can now be written as a linear combination of this conjugate basis:

$$e_0 = \underbrace{(\xi_1 d_1 + \dots + \xi_m d_m)}_{\in \mathbb{W}} + \underbrace{(\xi_{m+1} d_{m+1} + \dots + \xi_n d_n)}_{\in \mathbb{W}^{\perp A}} \quad (2.25)$$

The last step in the deflation technique is first to eliminate the components of the error that belong to the deflation space \mathbb{W} (i.e., $\{\xi_k\}_{k=1}^m$), and finally apply a CG-type algorithm to eliminate the remaining components of the error.

The first step (also known as the initial error projection step) is carried out by writing the initial error as:

$$e_0 = Wc_0 + e_1 \quad (2.26)$$

where W is a matrix whose columns are a basis of \mathbb{W} and c_0 is an m -dimensional vector. The vector c_0 can be easily determined noticing that by construction the vector e_1 is A -orthogonal with \mathbb{W} , i.e., $W^T A e_1 = 0$, which yields the following m -dimensional SPD system

of linear equations:

$$W^T A W c_0 = W^T r_0 \quad (2.27)$$

where $Ae_0 = r_0$ is the initial residual.

The initial guess is then updated as follows:

$$x_1 = x_0 + Wc \quad (2.28)$$

The last step is to apply a CG-type algorithm starting from the updated residual vector r_1 . Recall that the CG process generates a sequence of search directions from the residuals that are A -orthogonal to all the previous search directions (generated by the CG).

$$d_{k+1} = r_{k+1} + \beta_k d_k ; \quad \beta_k = \frac{(r_{k+1}, r_{k+1})}{(r_k, r_k)} \quad (2.29)$$

However, there is no guarantee that these search directions will belong to $\mathbb{W}^{\perp A}$. It is therefore necessary to include one more A -orthogonality condition to be satisfied by all the search directions:

$$W^T A d_k = 0 ; \quad \forall k \quad (2.30)$$

i.e., all the search directions need to be A -orthogonal to \mathbb{W} .

The new sequence of search directions can be constructed as follows:

$$d_{k+1} = r_{k+1} + \beta_k d_k - Wc_{k+1} \quad (2.31)$$

where the parameter β_k is determined by the A -orthogonality of successive search directions (as for the standard CG), and the m -dimensional vector c_{k+1} is determined by the A -orthogonality condition (2.30) between the search directions and \mathbb{W} . This yields the

following m -dimensional SPD system of linear equations:

$$W^T AWc_{k+1} = W^T Ar_{k+1} \quad (2.32)$$

Note that this new A -orthogonality conditions has to be also included when computing the first search direction from the updated initial residual.

The Deflated Conjugate Gradients algorithm reads as follows:

Algorithm 2.3.1 Deflated Conjugate Gradients (DCG)

- 1: Given A , b , W and an initial guess x_0
 - 2: Compute $r_0 := b - Ax_0$
 - 3: Solve $W^T AWc_0 = W^T r_0$
 - 4: Set $x_1 := x_0 + Wc_0$
 - 5: Compute $r_1 := b - Ax_1$
 - 6: Solve $W^T AWc_1 = W^T Ar_1$
 - 7: Set $d_1 := r_1 - Wc_1$
 - 8: **do until** convergence
 - 9: $\alpha_k := (r_k, r_k)/(d_k, Ad_k)$
 - 10: $x_{k+1} := x_k + \alpha_k d_k$
 - 11: $r_{k+1} := r_k - \alpha_k Ad_k$
 - 12: $\beta_k := (r_{k+1}, r_{k+1})/(r_k, r_k)$
 - 13: $W^T AWc_{k+1} = W^T Ar_{k+1}$
 - 14: $d_{k+1} := r_{k+1} + \beta_k d_k - Wc_{k+1}$
 - 15: **end do**
-

In terms of storage, the DCG algorithm requires four n -dimensional vectors (x , d , Ad and r), one m -dimensional vector (c), one m by n matrix (W) and one m by m matrix ($W^T AW$) to be saved. This extra storage can be mitigated when considering low dimensionality deflation spaces ($m \ll n$) together with high levels of sparsity in the W matrix. An examples of this is a subdomain deflation technique using constant shape functions. The solution/inversion of the direct system of equations (steps 3, 6 and 13 of algorithm 2.3.1) can be carried out very efficiently by performing a Cholesky factorization of the SPD matrix $W^T AW$ at the beginning of the algorithm, and a backward/forward substitution when required. As the number of non-zeros of $W^T AW$ can be very low, a skyline solver is the preferred choice to minimize the number of floating point operations. Preconditioning can be easily included in the same manner as for the standard CG method.

2.3.3 Deflation Subdomains

The DCG technique requires the definition of a deflation subspace \mathbb{W} . For matrices arising from finite element methods, the subdomain deflation approach[90] defines a deflation subspace based on a coarse discretization of the domain. This technique attempts to spatially approximate the smallest eigenmodes of the underlying system so that they can be removed from the search space of the conjugate gradients method.

It is well known that the discretization of the Laplace operator leads to ill-conditioned SPD matrices in cases of high aspect-ratio domains. This is usually due to the presence of very low eigenmodes pointing in the largest direction as shown in Figure 2.1.

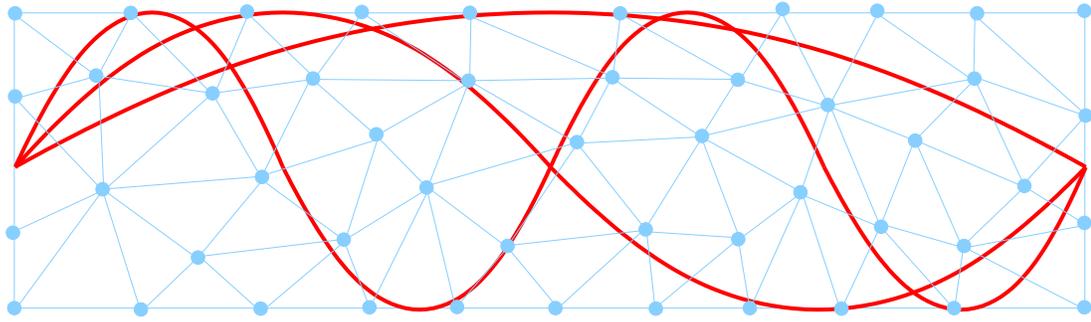


Figure 2.1: DPCG: Smallest eigenmodes of the Laplace operator on a rectangular domain.

A physical interpretation of this behavior is presented as follows. First note that values at the prescribed points have an influence over the whole domain due to the elliptic character of the Laplace operator. This implies, for example, that changes at opposite points along the longest direction will have to travel all along the domain. Therefore, if an iterative solver is used to solve the system of equations, the number of iterations in order to achieve convergence would be at least the number of elements that exists along the largest direction. This is due to the fact that each iteration (i.e., matrix/vector multiplication) only transmits information between neighboring nodes due to the local support of the interpolant functions used in the spatial discretization of the domain.

This suggests that, for these cases, a good strategy for approximating the smallest eigenmodes of the system would involve a coarse discretization oriented along the largest

direction of the domain. One of the simplest ways of defining a coarser discretization from a given unstructured mesh is to agglomerate the nodes of the mesh into subdomains. Two alternative methods have been developed following this strategy:

- Seedpoints Alternative (SA): For this (manual) technique, the user defines an arbitrary set of points, called *seedpoints*. Given a mesh, the closest mesh points to the seedpoints are found, and a region number is assigned accordingly. Points not assigned to any region are then added one layer at a time until all points have been assigned a region number.
- Advancing Layers Method (ALM): Starting from a prescribed point, neighboring points are added one layer at a time, until a specified number of points per region is exceeded. The last set of points added is then used as a starting point for the next group. The procedure is repeated until all points have been assigned a region number.

Note that in cases with very elongated domains, where the prescribed values are located at the extremum points, the ALM method will generate a coarse discretization that will be approximately aligned with the largest direction (provided that the number of points per region is sufficiently large). Note also that in contrast to the SA method, the ALM method does not require the intervention of the user for the generation of the subdomains (besides the specification of the number of points per region). However, the SA method provides a more flexible option when dealing with domains that does not clearly satisfy the elongated shape assumption. It is worth to mention that the prescribed points are not assigned to any region, since they do not belong to the solution space.

The last step is to define an interpolant polynomial over the deflation regions. The simplest choice are the constant shape-functions. Using this approach, each column of the matrix W will represent one deflation subdomain. For a given column, a unity value will

be assigned if a point belongs to that region, and zero otherwise.

$$W_{ij} = \begin{cases} 1 & p_i \in g_j \\ 0 & p_i \notin g_j \end{cases} \quad (2.33)$$

2.4 Benchmarks

The deflated PCG solver has been tested on a variety of examples. These 'test cases' can be divided in two groups: 1) Classic Benchmarks and 2) Hemodynamics Models. The aim of these examples was the comparison of the number of iterations of the pressure Poisson solver and the speed of the flow calculation. For all the cases, it was verified that the results obtained using the deflated PCG solver coincide with those obtained by the standard PCG solver. In addition, the results obtained in the benchmark tests agreed with those reported in the literature[95].

2.4.1 Pipe Flow

The first example is the classic Poiseuille pipe flow, a steady flow of a viscous Newtonian fluid in a straight circular domain. A uniform velocity profile is prescribed at the inflow, while a constant pressure is prescribed at the outflow. Since the pressure field has to establish itself along the pipe, the number of iterations required increases with the graph depth of the finite element mesh. The physical dimensions and parameters were set as follows:

- pipe radius: $r = 1$
- pipe length: $l = 20, 40, 80$
- density: $\rho = 1$
- inflow velocity: $v = 1$
- viscosity: $\mu = 0.01$

The element size was set to $h = 0.1$, implying approximately a graph depth of 200, 400 and 800 for the cases considered. This resulted in grids of 129 Kels, 260 Kels and 516 Kels respectively. All cases were run for 100 timesteps using explicit timestepping. The number of groups were chosen to be 15, 30 and 60. The deflation domains were generated by the advancing layers technique, starting from the exit. Figure 2.2-2.3 show the surface mesh, deflation domain boundaries for 15 groups, pressure and absolute value of the velocity for $l = 20$. As expected the velocity profile at the outlet is parabolic, and the pressure field is linear along the cylinder axis after an initial development distance.

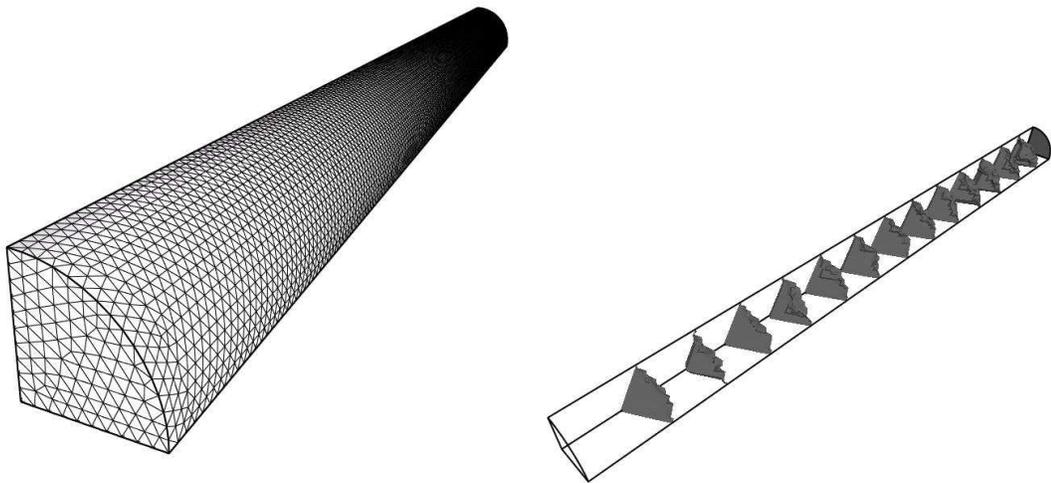


Figure 2.2: Pipe Flow: Surface mesh and deflation domain boundaries for $l = 20$

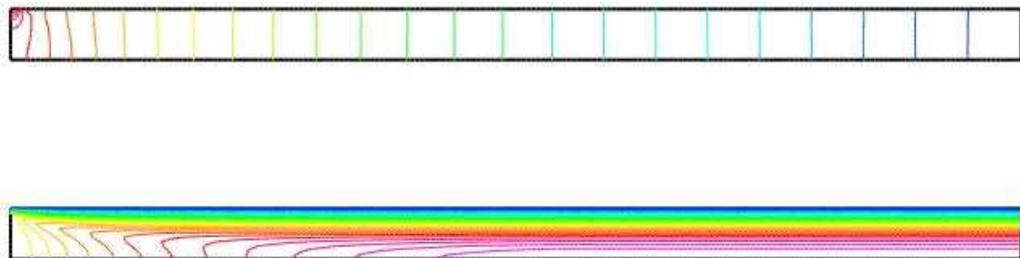


Figure 2.3: Pipe Flow: Pressure and Abs(Velocity) for plane $z = 0$.

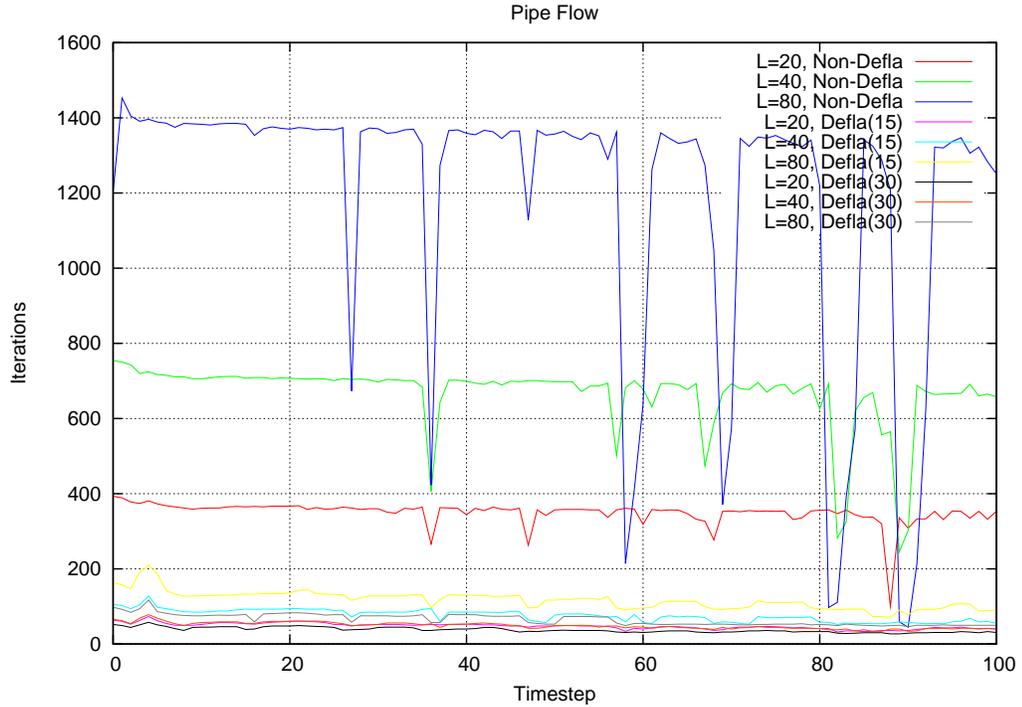


Figure 2.4: Pipe Flow: Number of iterations required for the PCG solver at each timestep.

Figure 2.4 shows the number of iterations required for the PCG solver. The sudden 'dips' in the number of iterations at some timesteps are due to the fact that a projective prediction of pressure increments with 2 Krylov vectors [76] was used. Note the dramatic decrease in the number of iterations achieved by the deflated PCG solver. This decrease may also be seen in Figure 2.5, which depicts the average number of iterations for the first 20 steps for the different options chosen.

While the number of iterations increases linearly with the pipe length for the conventional PCG, the performance of the deflated PCG solver seems to be insensitive to the pipe length and the number of groups chosen. Figure 2.6 shows the total CPU time required for the simulation, highlighting the importance of a fast Pressure-Poisson solver. Note that for the case $l = 80$, the deflated PCG case performs seven times faster.

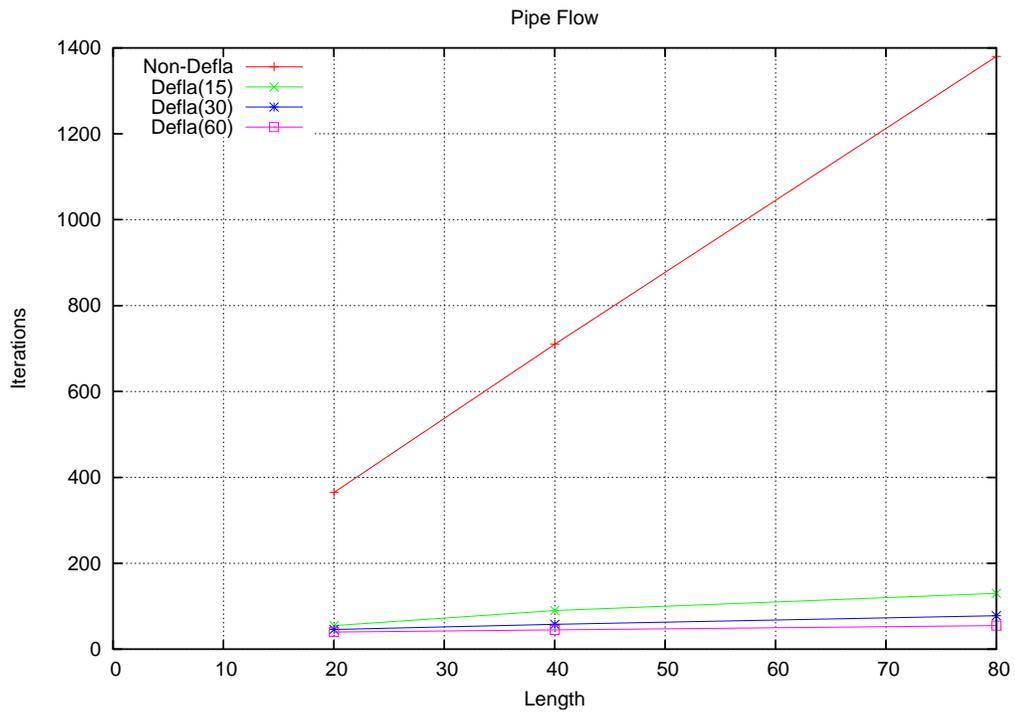


Figure 2.5: Pipe Flow: Average number of iterations required for the PCG solver as a function of the pipe length.

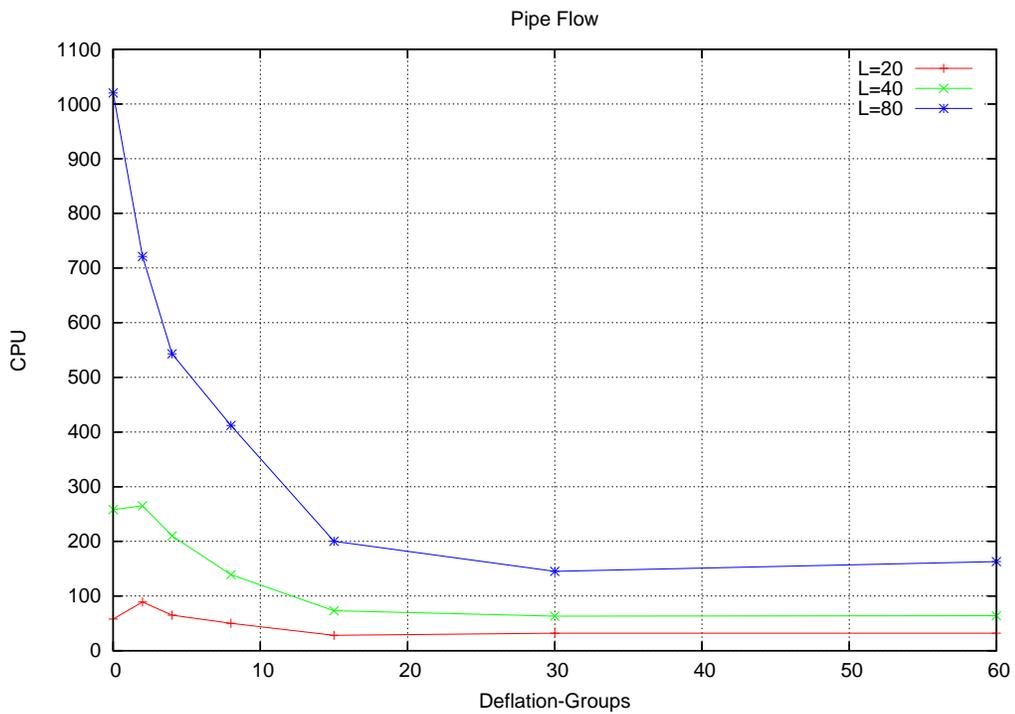


Figure 2.6: Pipe Flow: Total CPU time required for 100 timesteps.

2.4.2 Naca 0012

The second example is the classic NACA0012 airfoil at $\alpha = 5^\circ$ angle of attack. This is a steady, inviscid case (Euler equations). Figures 2.7 show the surface mesh employed, as well as the surface pressures obtained.

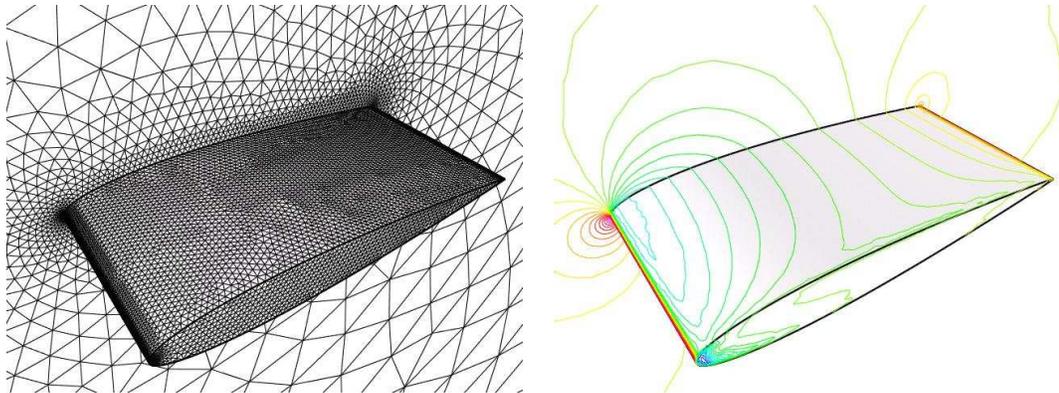


Figure 2.7: NACA 0012: Surface Mesh and Pressure contours.

The mesh had approximately 370 Kels. This problem was solved using local timesteps to accelerate convergence to steady-state. The deflation subdomains were generated from 39 seedpoints selected by hand. The seedpoints are shown in figure 2.8

Figure 2.9 shows the number of iterations as the solution is advanced to steady state. One can observe that even on this rather coarse mesh with limited graph depth between the outflow, prescribed pressure boundary and the airfoil, the deflated PCG requires approximately half the iterations of the usual PCG. In this case only a modest improvement of performance was obtained.

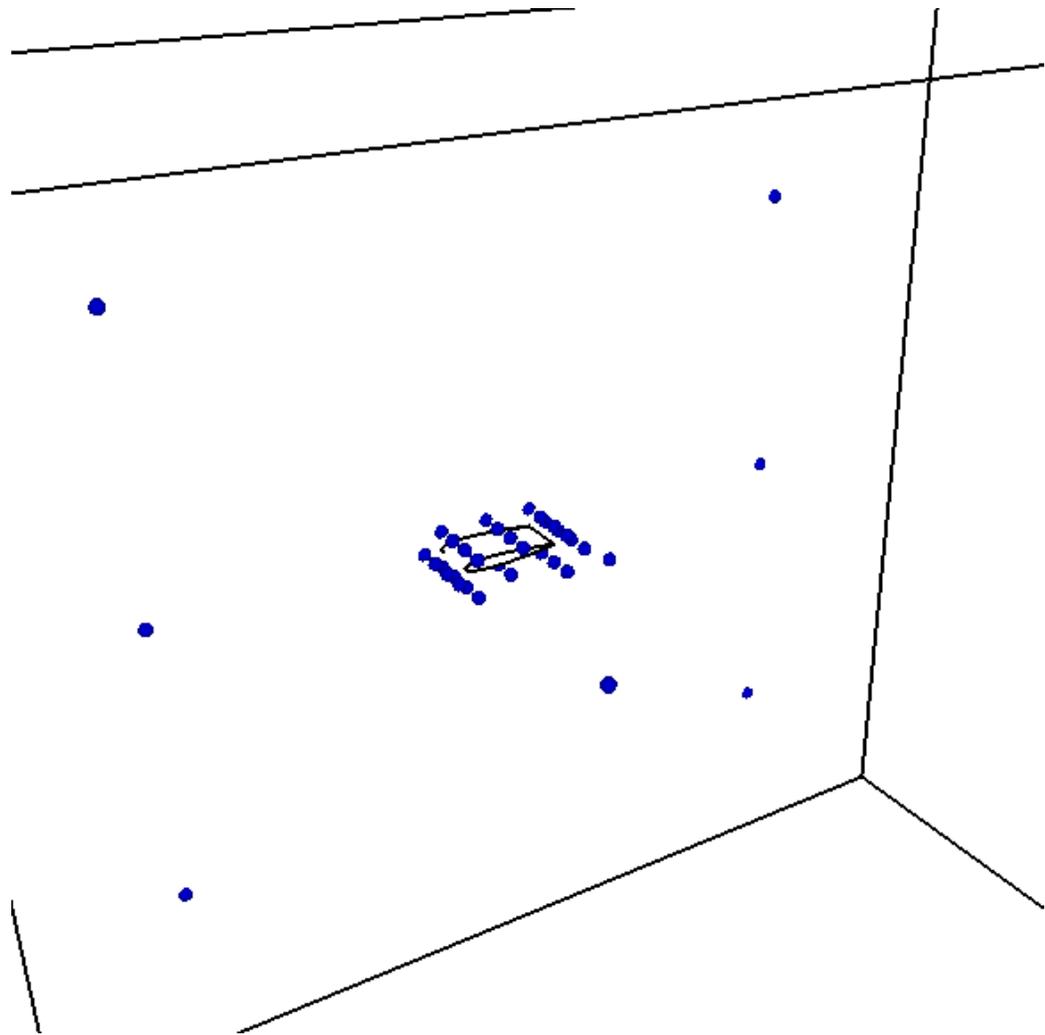


Figure 2.8: NACA 0012: Seedpoints selected by hand used to generate 39 deflation subdomains.

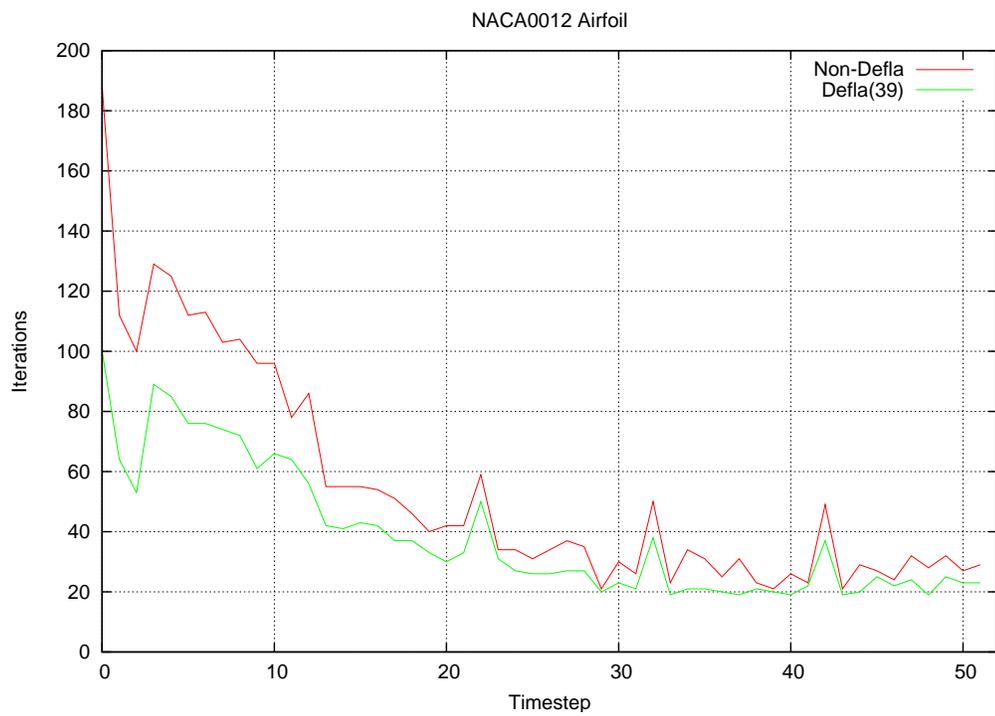


Figure 2.9: NACA 0012: Number of iterations per pseudo timestep required to achieve steady state.

2.4.3 von Karman Vortex Street

The last example in this section is also a well known benchmark case. A circular cylinder is suspended in a uniform stream of incompressible viscous fluid. The separation at the back of the cylinder generates the so-called von Karman vortex street, whose characteristics depend on the Reynolds number $Re = \rho V_\infty D / \mu$, where D is the diameter of the cylinder. This is essentially a 2-D example, but it was run with a 3-D solver. A mesh of 113 Kels was used for the simulation, with special placement of points in the vicinity of the cylinder. The parameters were chosen such that the resulting Reynolds number was $Re = 190$.

Figure 2.10 shows the surface grid of the 3-D model. Figure 2.11 shows the surface grid and the absolute value of the velocity in a cut plane.

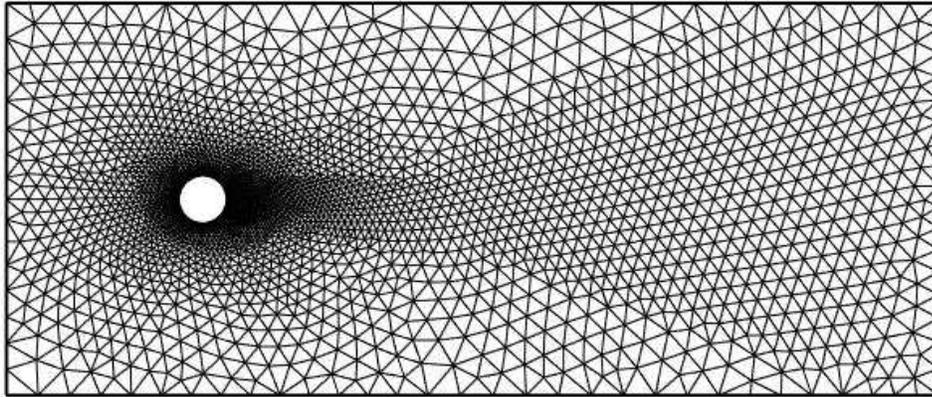


Figure 2.10: von Karman Vortex Street: Surface mesh of the 3-D model.

The run was started impulsively and continued until the vortex street was fully developed. Starting from this (restart) state, the solution was advanced 50 steps using a 3-stage Runge-Kutta scheme for the advection, and two different options of deflation sub-domains were tested. In both cases the deflated PCG regions were grown from 14 and 18 manually selected seedpoints. Figure 2.12 shows the deflation regions for the 18 seedpoints case.

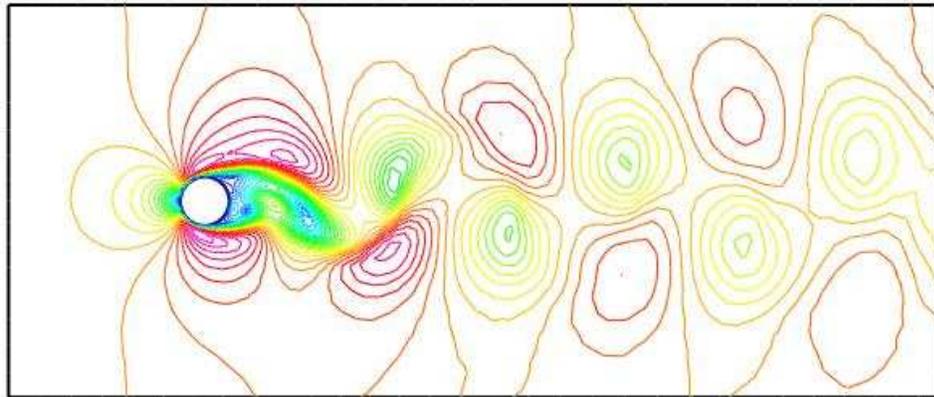
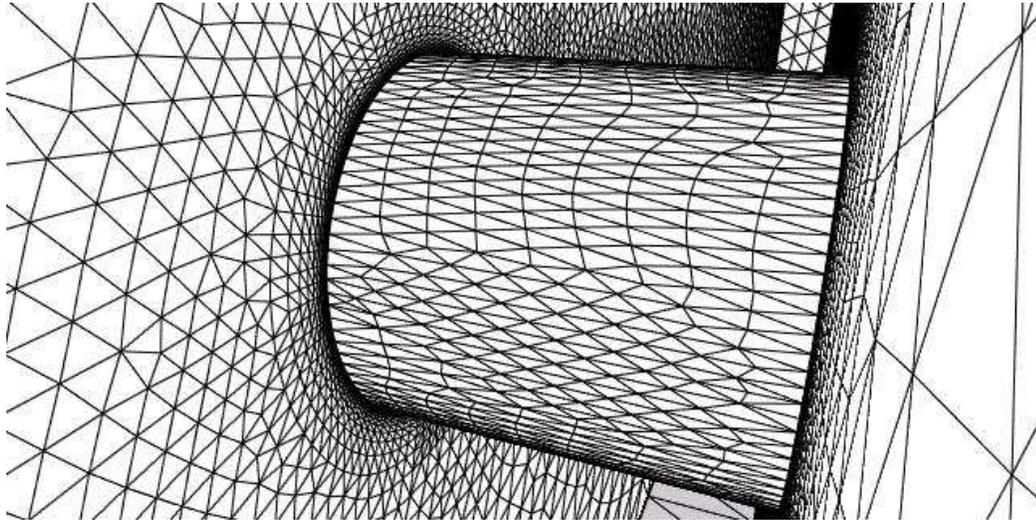


Figure 2.11: von Karman Vortex Street: Surface mesh and Abs(Vel).

The iterations required per timestep are displayed in Figure 2.13. One can observe that the deflated PCG requires substantially less iterations, and is rather insensitive to the number of subdomains chosen. Note that in this case where the grid is highly stretched around the cylinder (Navier-Stokes mesh), the deflated PCG still reduced the number of iterations.

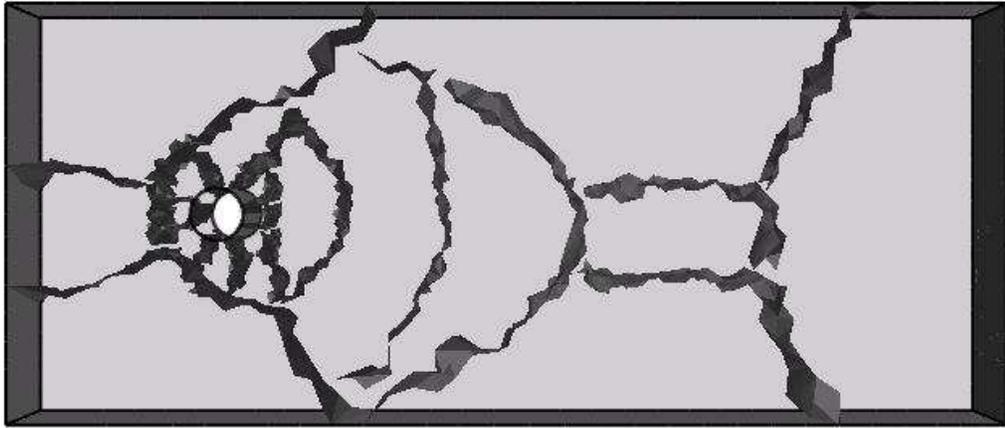


Figure 2.12: von Karman Vortex Street: Deflation subdomains grown from 18 seedpoints.

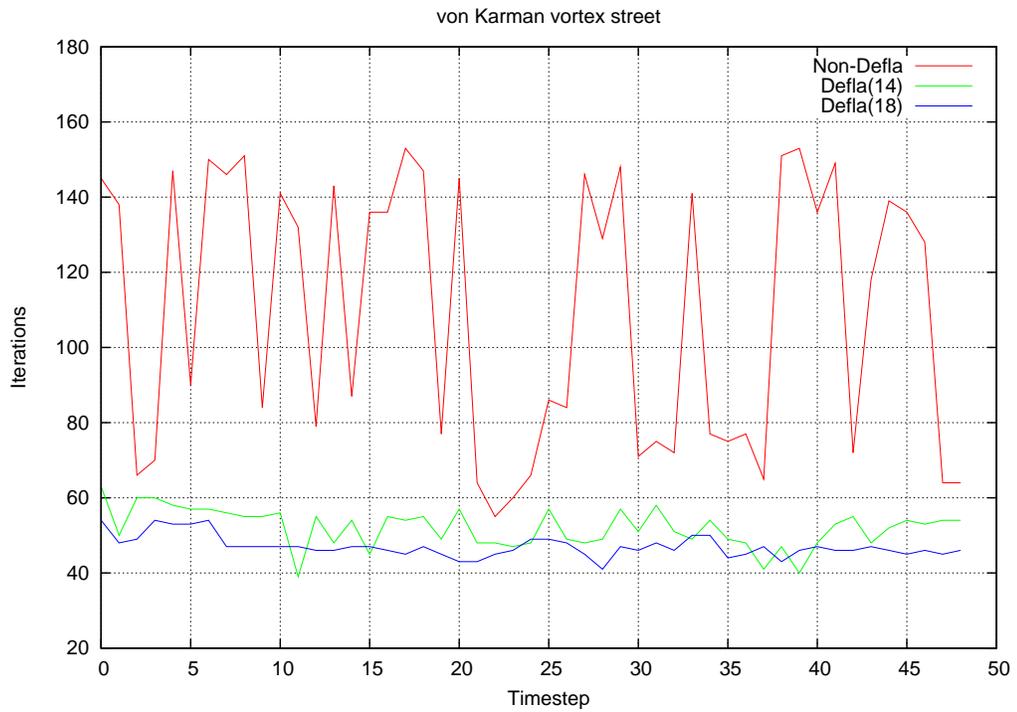


Figure 2.13: von Karman Vortex Street: Number of iterations per timestep.

2.5 Hemodynamic Models

The deflated PCG has been tested on five different patient-specific vascular models to assess its performance in real hemodynamics cases. Each case is representative of the different geometries encountered when constructing patient-specific models of aneurysms and cerebral arteries. For all the models, the following assumptions were made.

The blood was considered as a Newtonian fluid, which is modeled by the *unsteady* incompressible Navier-Stokes equations. The simulations were performed using implicit timestepping, solving a pseudo-steady problem at each timestep[80]. Within each pseudo-timestep the advective terms were integrated implicitly using 5 LU-SGS passes (local Courant number $C = 5.0$), followed by the pressure projection. The timestep was set to $\Delta t = 0.01$ sec. The material properties of the fluid were taken to be $\rho = 1.0$ g/cm³ and $\mu = 0.04$ Poise. Two cardiac cycles were computed with 100 timesteps per cycle. The first three examples were computed on an Intel Xeon processor (E5345/2.33MHz) computer with 16GB of RAM using the serial version of the flow solver. The last example (cerebral arteries) was run on an 8-way Dual-Core AMD Opteron processor (8222/3MHz) machine with 64GB of RAM using the OpenMP version of the flow solver on 8 Processors.

2.5.1 Internal Carotid Artery Aneurysm

The first example is a patient-specific model of a cerebral aneurysm located at the Internal Carotid Artery (ICA). This case is particularly interesting since it was possible to reconstruct the entire ICA from the carotid bifurcation to the carotid terminals from 3DRA data. This model was previously used in a study[96] to investigate how much of the upstream vessel has to be included in the model in order to obtain the right secondary flows at the aneurysm. The volume mesh obtained for this model had 646 Kpts and 3.6 Mels. The deflation groups were generated using both methods. For the seedpoints alternative 45 points were manually selected from the surface model. For the advancing layers method 20, 50, 100 and 150 groups were automatically generated. Figure 2.14 shows the overall domain,

as well as the group boundaries for the 45 and 50 groups cases.



Figure 2.14: ICA Aneurysm: Deflation boundary subdomains for 50 groups using advancing layers method (left) and 45 groups using the seedpoints alternative (right).

The pressure was prescribed (homogeneous bc) at the outflow boundary at the ICA terminal (top part), while a time-dependent velocity profile was prescribed at the inflow boundary at the origin of the ICA (bottom part). No-slip boundary conditions were applied at the vessel wall.

Figure 2.15 depicts the pressure drop (left) and the wall shear stress distribution (right) at the peak systole (the inflow rate peak) of the second cardiac cycle. As it was expected, the pressure gradient is aligned with the parent vessel, while the pressure is almost constant inside the aneurysm.

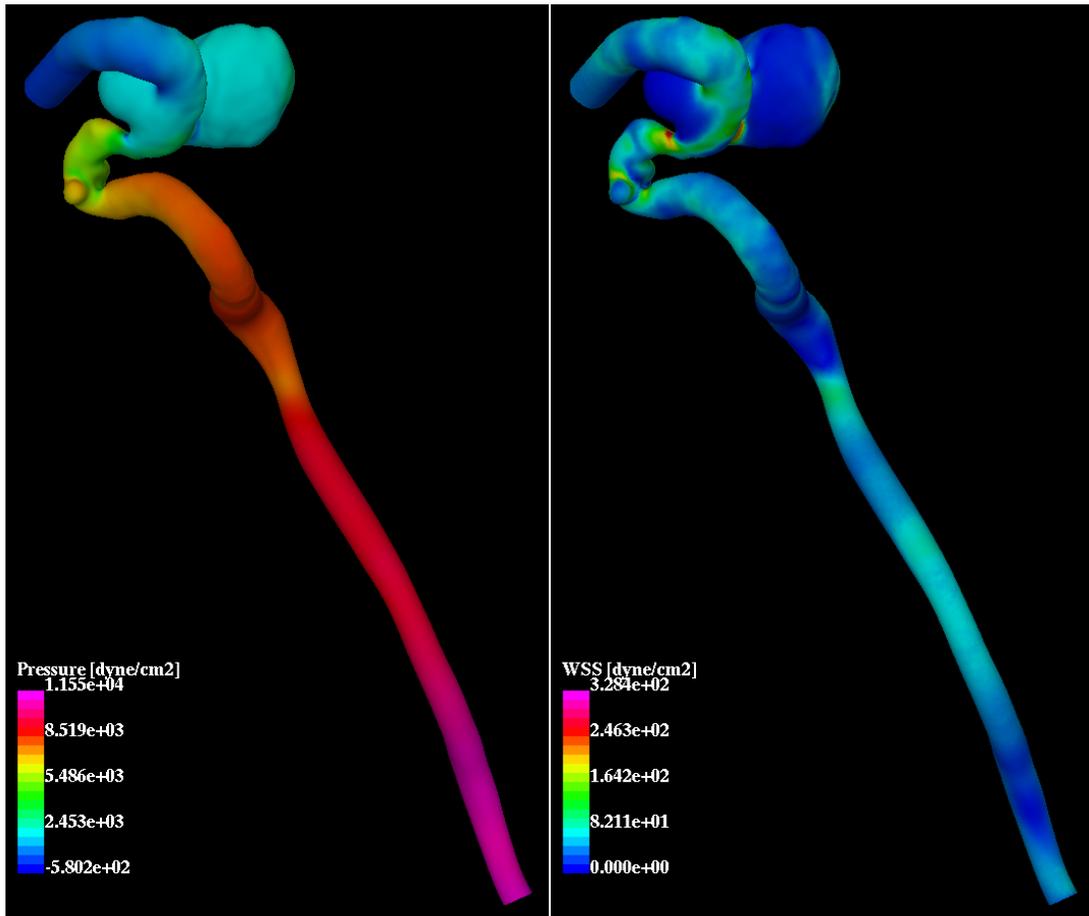


Figure 2.15: ICA Aneurysm: Pressure (left) and Wall Shear Stress (right) distributions. at $T = 1.4\text{sec}$.

Figure 2.16 shows the average number of iterations required for the pressure Poisson solver at each timestep. This average was computed over all the iterations that took place during the several pseudo timesteps within each timestep. The average number of iterations without deflation has a mean value of 625, while with deflation it ranges between 75 and 150 iterations. Therefore, in average, the number of iterations have decreased between 4.20 and 8.45 times for the cases with 20 and 150 groups, respectively.

Figure 2.17 depicts the CPU time required to complete each timestep. For the non-deflated case, the average CPU time of the whole run is about 15m53s, while with deflation

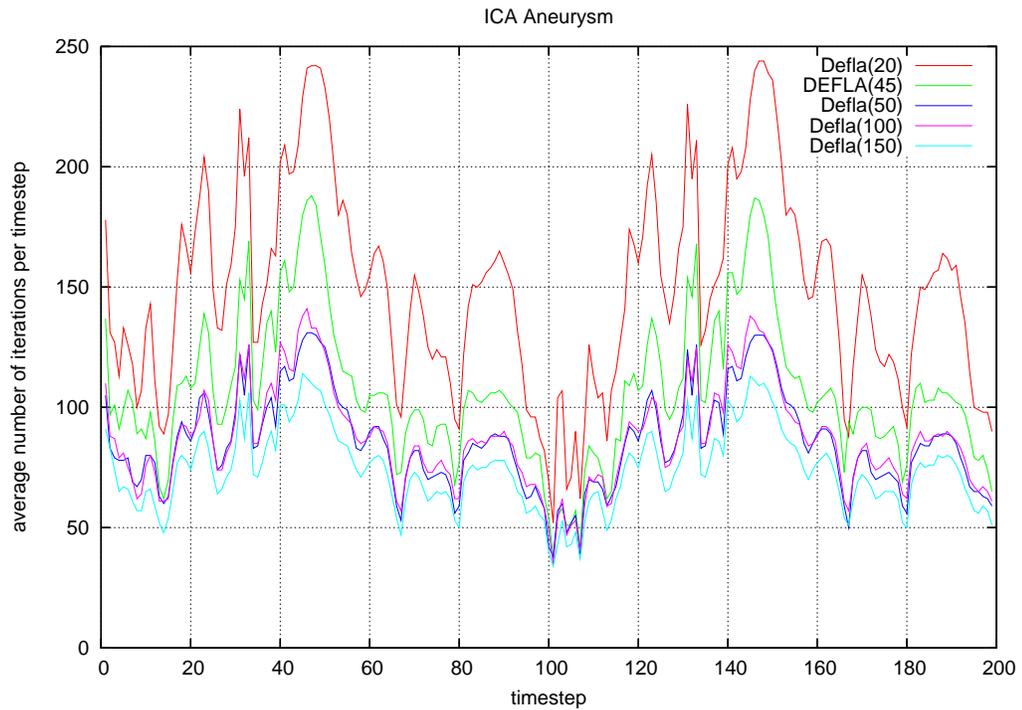
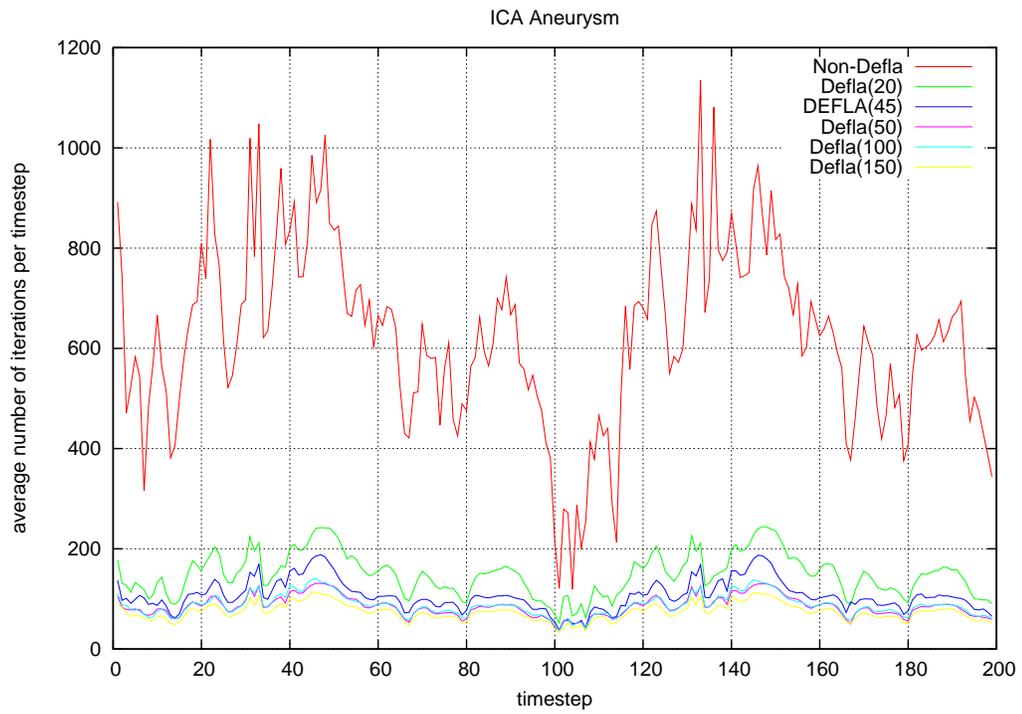


Figure 2.16: ICA Aneurysm: Average number of iterations for the Pressure-Poisson solver at each timestep for all cases (top) and deflation cases only (bottom).

it ranges between 4m55s and 6m40s seconds. This implies that the average speedup is between 2.4 and 3.2 for the cases with 20 and 150 groups respectively. Looking at the deflation cases only (bottom part of both figures), it is clear that as the number of deflation groups increases, the average number of iterations and the CPU time per timestep always decreases. By comparing the 45 groups case (seedpoints) and the 50 groups case (advancing layers), it can be concluded that both methods have performed similarly in this case.

Looking at both figures, it can be seen that all the curves follow the inflow profile, having the highest values where the inflow rate is maximum. This implies that for a higher inflow rate, more iterations are needed by the pressure Poisson solve in order to reach convergence. The sudden 'dips' are due to a good approximation of the initial guess using the solution of the previous timestep. This behavior is accentuated at the end of the cardiac cycles (1st and 2nd) where the inflow rate changes very slowly.

Figure 2.18 shows the total CPU time for all the cases run. The best speedup obtained is about 3.18 with 150 groups.

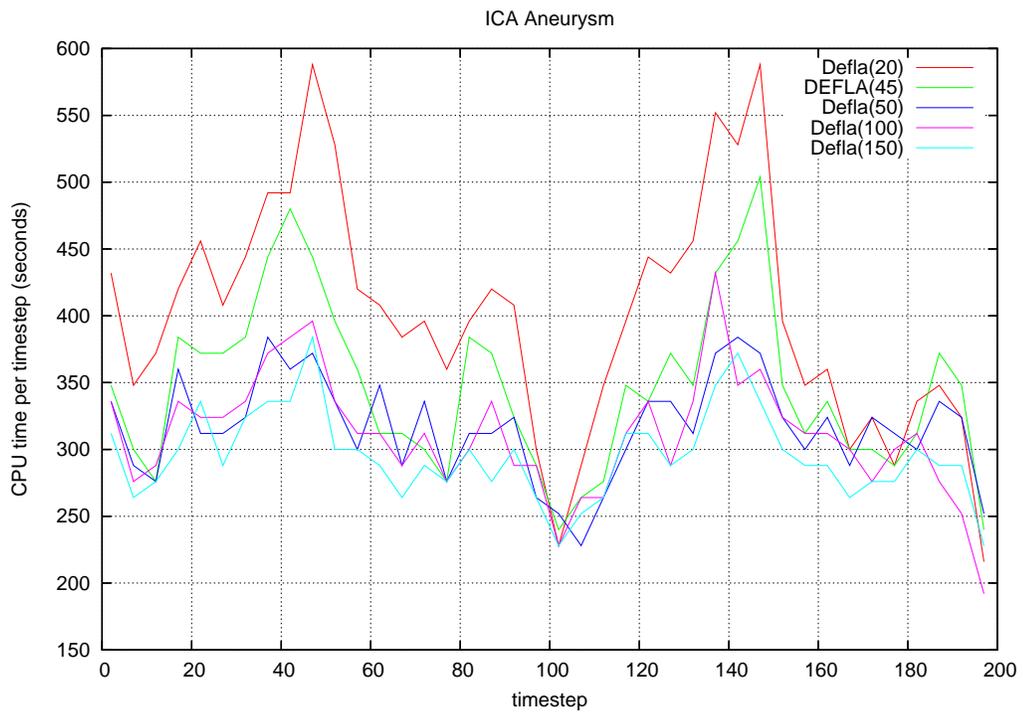
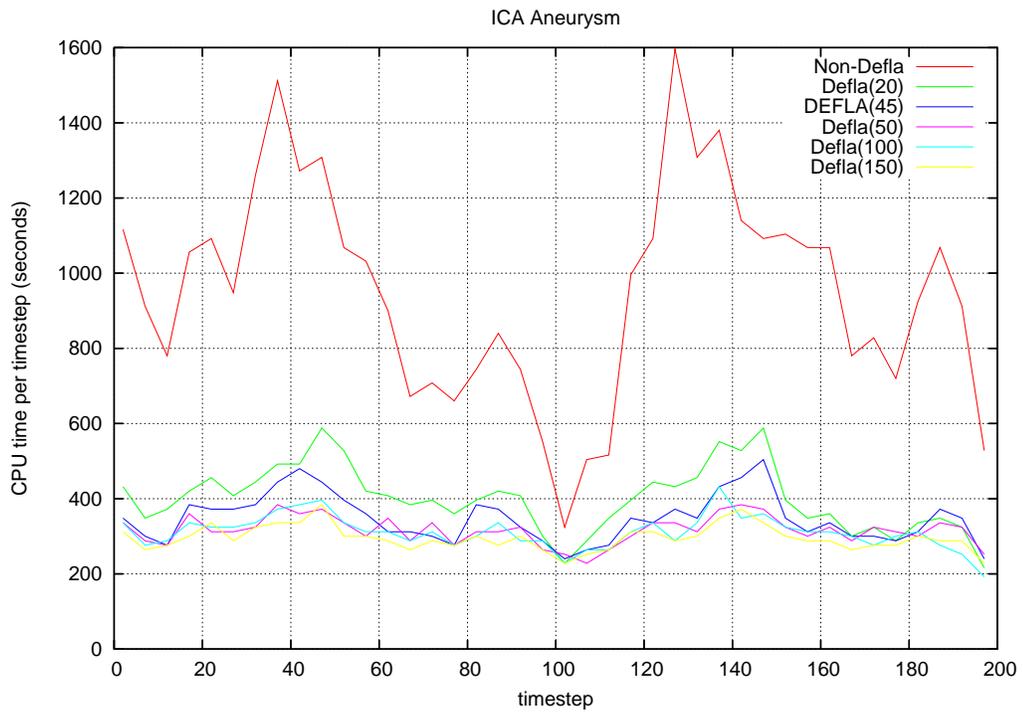


Figure 2.17: ICA Aneurysm: CPU time at each timestep for all cases (top) and deflation cases only (bottom).

ICA Aneurysm - Timings

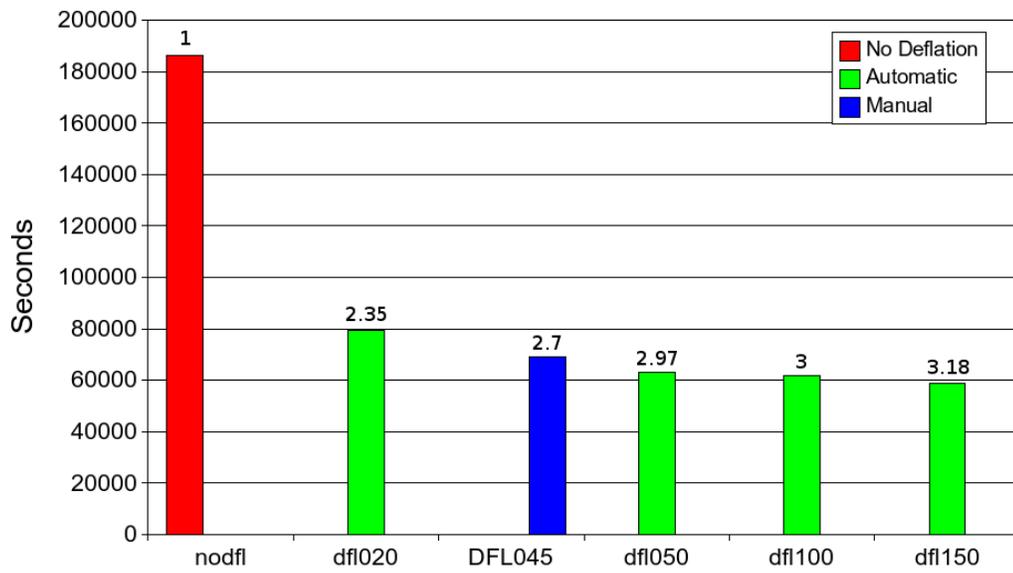


Figure 2.18: ICA Aneurysm: Total CPU time required for the 200 timesteps. The best speedup obtained was about 3.18 corresponding to 150 groups using the advancing layers technique.

2.5.2 Basilar Tip Aneurysm

The second example is a patient-specific model of a cerebral aneurysm located at the tip of the basilar artery. The model included in the upstream direction both vertebral arteries. The volume mesh generated for this model had 535 Kpts and 2.9 Mels. The deflation subdomains were generated using both methods. For the seedpoints alternative 90 points were manually selected from the surface model. For the advancing layers method 20, 50, 100 and 150 groups were automatically generated. Figure 2.19 shows the overall domain, as well as the deflation group boundaries for the 90 and 100 groups cases.



Figure 2.19: BT Aneurysm: Deflation boundary subdomains for 100 groups using the advancing layers method (left) and 90 groups using the seedpoints alternative (right).

The pressure was prescribed (homogeneous bc) at the three outflows (top part), while a time-dependent velocity profile was prescribed at the two inflows (bottom part). No-slip boundary conditions were applied at the vessel wall.

Figure 2.20 depicts the pressure (left) and the wall shear stress distribution (right) at the peak systole of the second cardiac cycle. As it was expected, the pressure changes along the vessels, while it is almost constant inside the aneurysm.

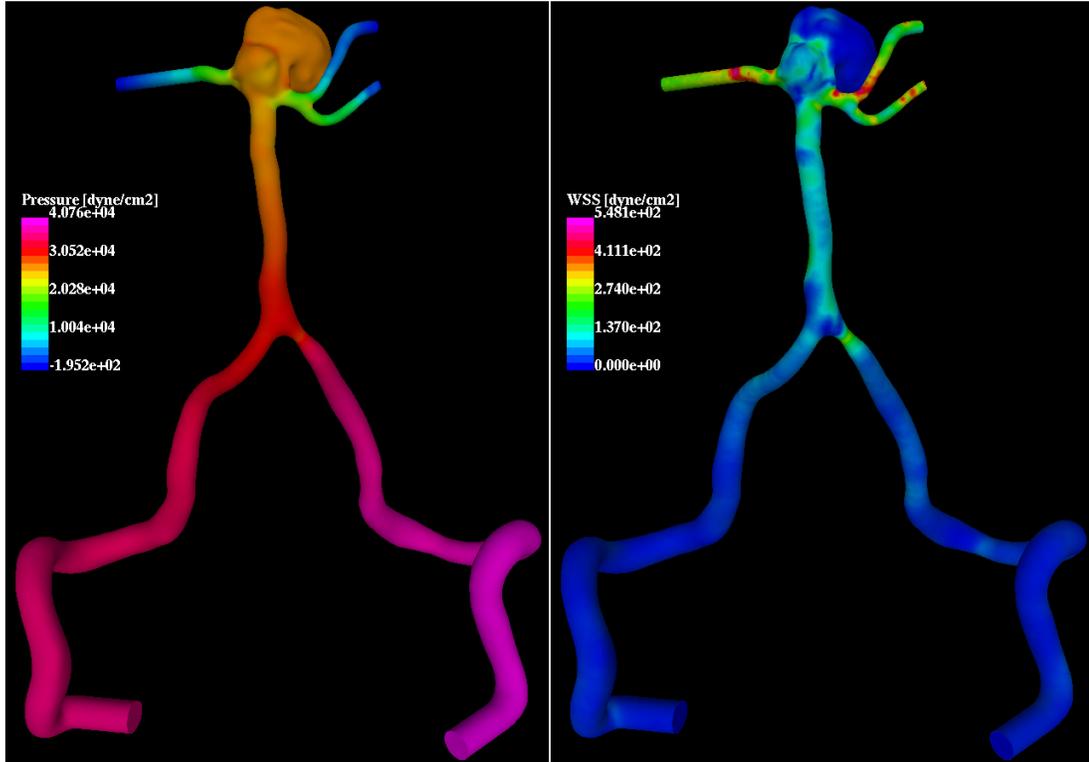


Figure 2.20: BT Aneurysm: Pressure and Wall Shear Stress distribution at $T = 1.4\text{sec}$.

Figure 2.21 shows the average number of iterations required for the pressure Poisson solver at each timestep. For the standard PCG solver, the average number of iterations have a mean value of 820 iterations. For the deflated PCG solver, the mean values of the average number of iterations range from 50 to 190 iterations. This implies that in average, the number of iterations have decreased between 4.2 and 15.7 times corresponding to the 20 groups and 90 groups cases respectively. Note also that for the non-deflated case the standard PCG solver have not reached convergence as it reached the maximum pre-set limit of 1200 iterations in several timesteps.

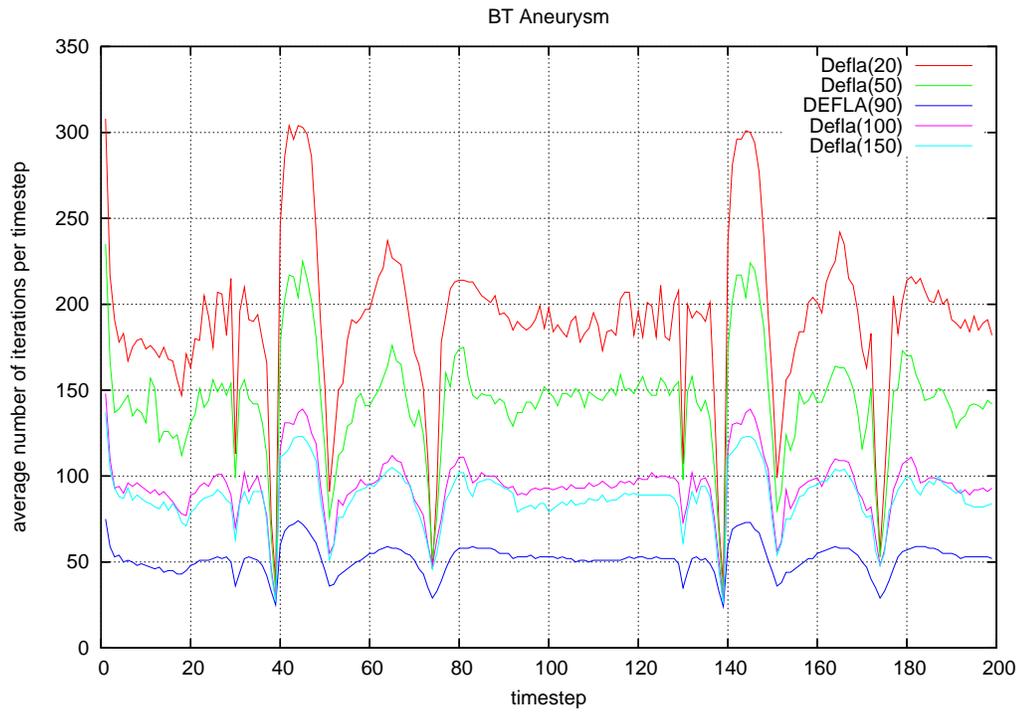
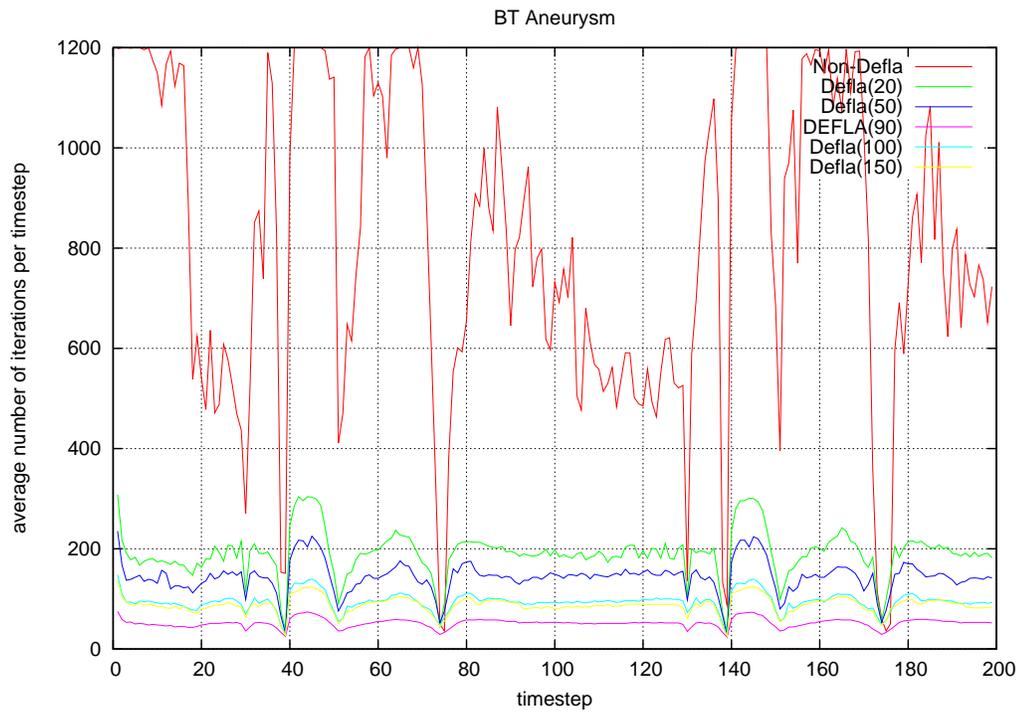


Figure 2.21: BT Aneurysm: Average number of iterations for the Pressure-Poisson solver at each timestep for all the cases (top) and the deflation cases only (bottom).

Figure 2.22 depict the CPU time required to complete each timestep. For the non deflated case the average CPU time is about 14 minutes, while for the deflated cases this value ranges from 3m20s and 6m10s. This implies that the average speedup of the whole run is between 2.38 and 4.21 for the cases with 20 and 90 groups, respectively. The best speedup obtained corresponds to the case with 90 groups using the seedpoints alternative, which performed even better than the 150 groups using the advancing layers method. This may be due to the fact that with the seedpoints alternative the resulting groups have more regular shapes than those generated by the ALM method as observed in figure 2.19 As in the previous example, the sudden 'dips' found in both figures can be attributed to a good approximation of the initial guess using the solution of the previous timestep.

Figure 2.23 shows the total CPU time for all the cases run. The best speedup obtained is about 4.23 with 90 groups.

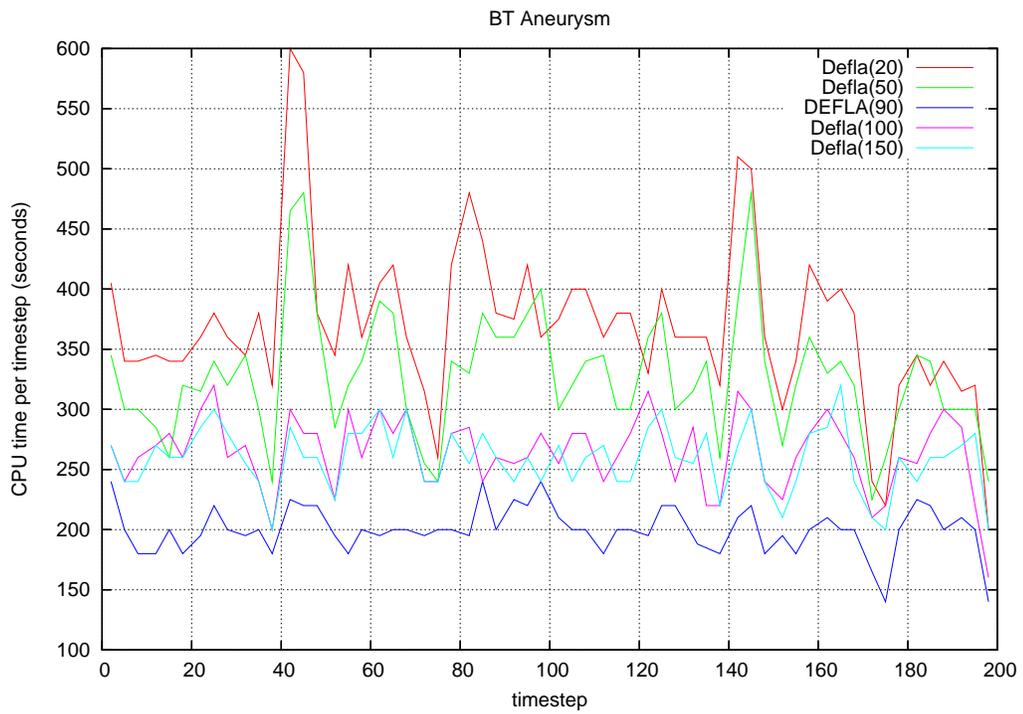
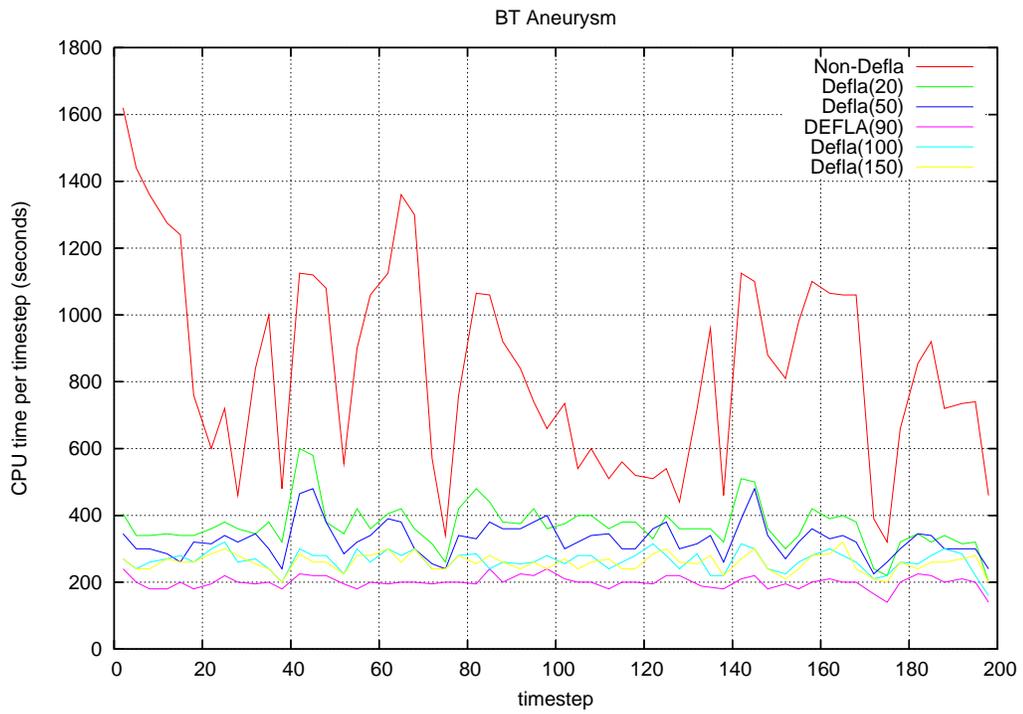


Figure 2.22: BT Aneurysm: CPU time at each timestep corresponding to all the cases (top) and the deflation cases only (bottom).

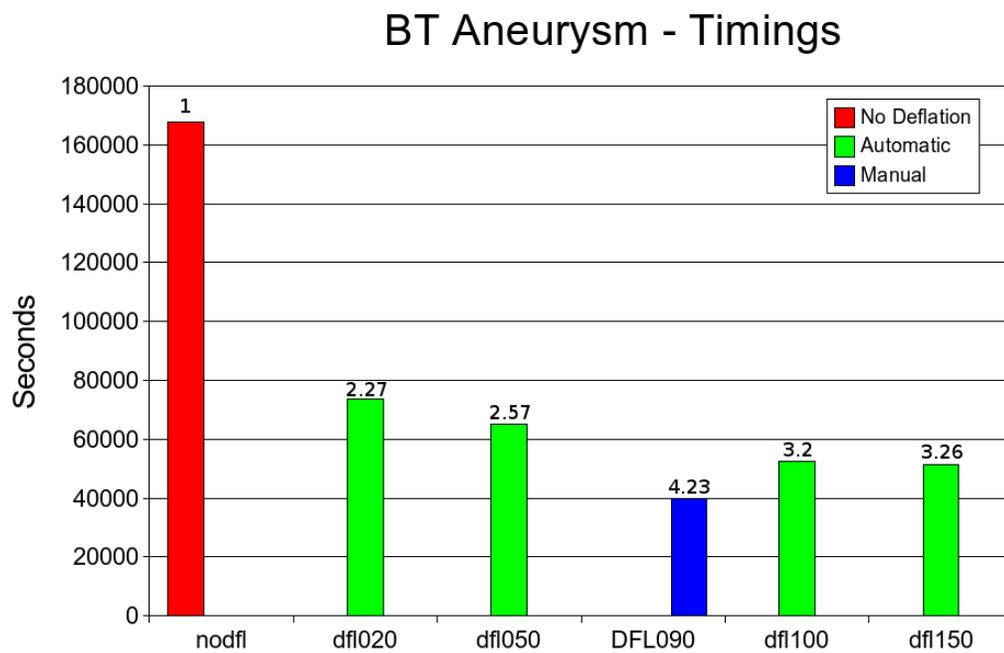


Figure 2.23: BT Aneurysm: Timings corresponding to the basilar model of a cerebral aneurysm. The best speedup obtained was about 4.23 corresponding to 90 groups using the seedpoints alternative.

2.5.3 Circle of Willis

The third example is a subject-specific model of the circle the Willis of a normal volunteer constructed from magnetic resonance images. The volume mesh generated for this model had approximately 836 Kpts and 4.6 Mels. The deflation groups were generated using both methods. For the seedpoints alternative 131 points were manually selected from the surface of the model. For the advancing layers method 20, 50, 100 and 150 groups were automatically generated. Figure 2.24 shows the overall domain, as well as the deflation subdomain boundaries for the 131 and 150 groups cases.

The pressure was prescribed (homogeneous bc) at the eight outflows (left and right ICAs and basilar artery). Time-dependent velocity profiles constructed from patient-specific flow rates obtained from PC MR images were prescribed at the three inflows (bottom part).

Figure 2.25 depicts the pressure (top) and the wall shear stress distribution (bottom) at the peak systole of the second cardiac cycle. As it can be seen in this picture, the pressure gradient is very low on the large vessels when compared with the small vessels. The higher pressure gradients are located in the places where the vessel's radius changes more rapidly. This is linked to the fact that in those regions the resistance to the blood flow increases considerably. These changes in the resistance are also noticeable by the wall shear stress distribution, which is also higher in the regions where the vessel's radius decreases more rapidly.

Figure 2.26 shows the average number of iterations required for the pressure Poisson solver at each timestep. For the standard PCG solver, the average number of iterations have a mean value of 860 iterations. For the deflated PCG solver, the mean values range from 90 and 190 iterations. This implies that, in average, the number of iterations have decreased a factor between 2.3 and 9.8 for the 20 groups case and the 131 groups case respectively.

Figure 2.27 depicts the CPU time required to complete each timestep. For the non-deflated case the average CPU time is about 23 minutes, while for the deflated cases this values ranges from 7m30s and 14m50s. The average speedup is therefore between 1.6 and 3 for the cases with 20 and 131 groups, respectively. Note that for the deflated cases using the advancing layers method, as the number of groups increases, the average number of iterations and the CPU-time per timestep decreases accordingly. The best speedup in both iterations and CPU time was obtained for the 131 groups using the seedpoints alternative. As in the previous case this seems to be related to a more uniform groups shape using the seedpoints alternative when compared to the advancing layers method.

Figure 2.28 shows the total CPU time for all the cases run. The best speedup obtained is about 3.9 with 131 groups.

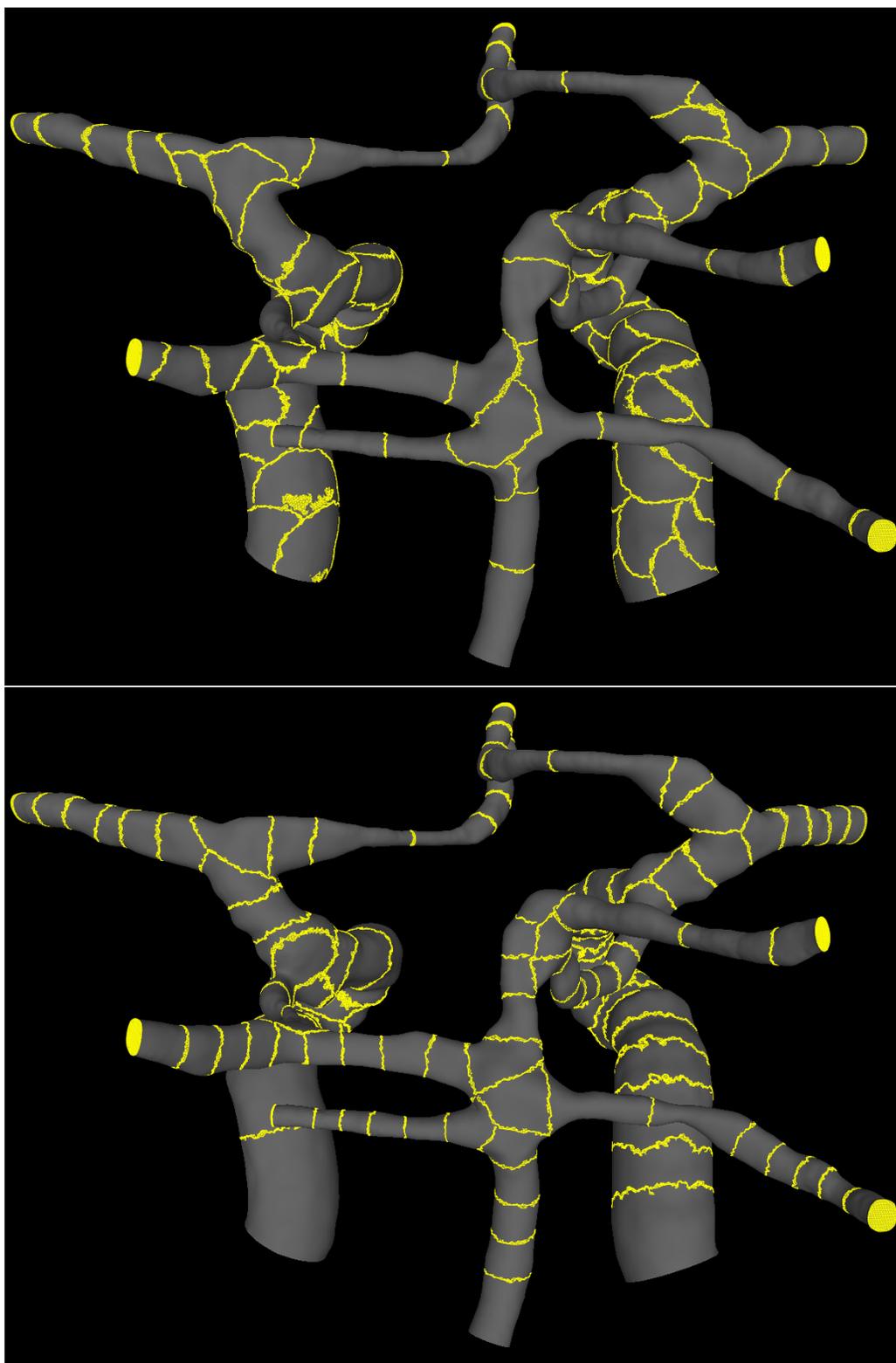


Figure 2.24: COW: Deflation boundary subdomains for 150 groups using advancing layers method (left) and 131 groups using the seedpoints alternative (right).

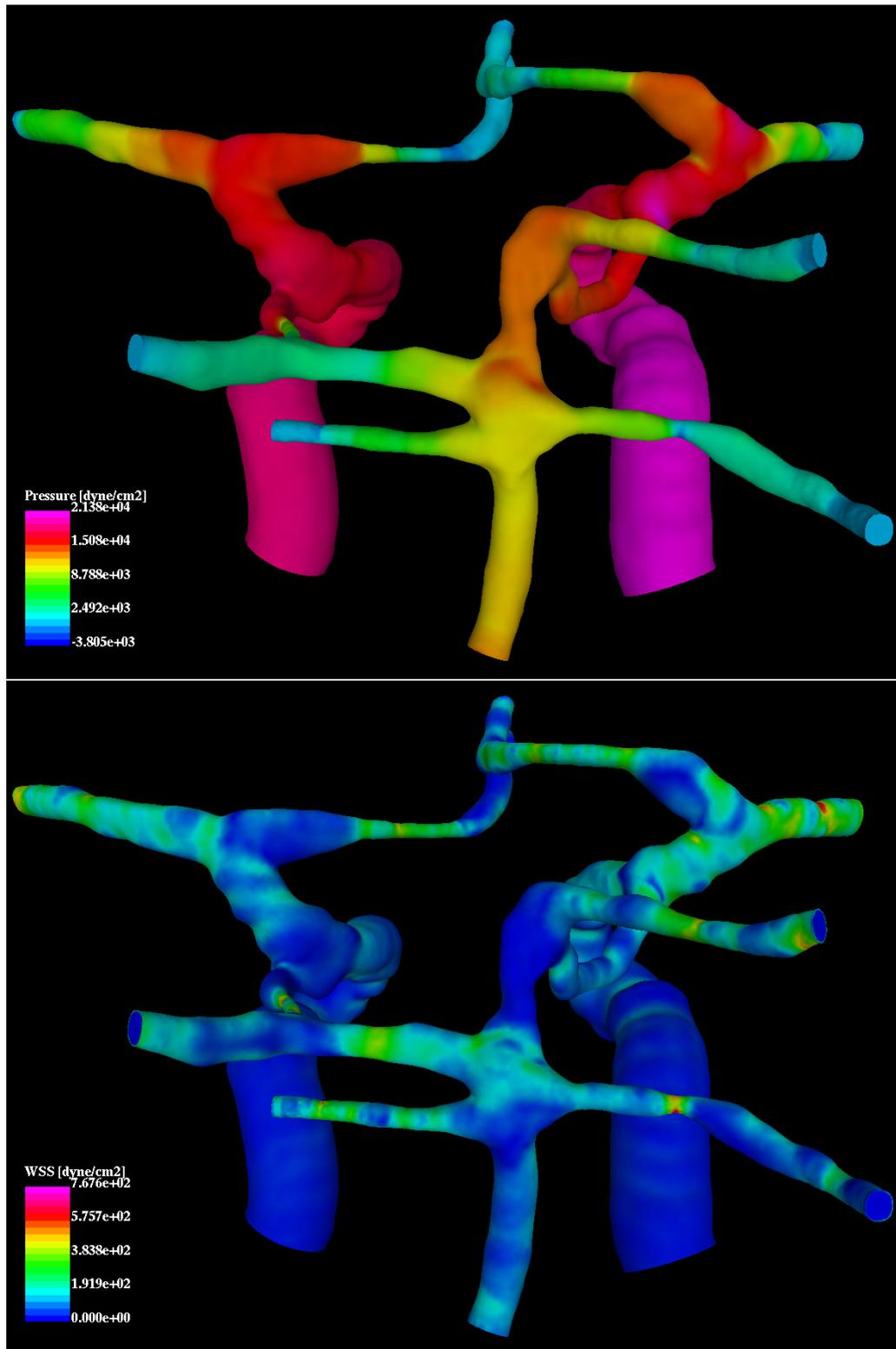


Figure 2.25: COW: Pressure and Wall Shear Stress distributions at $T = 1.4$ sec.

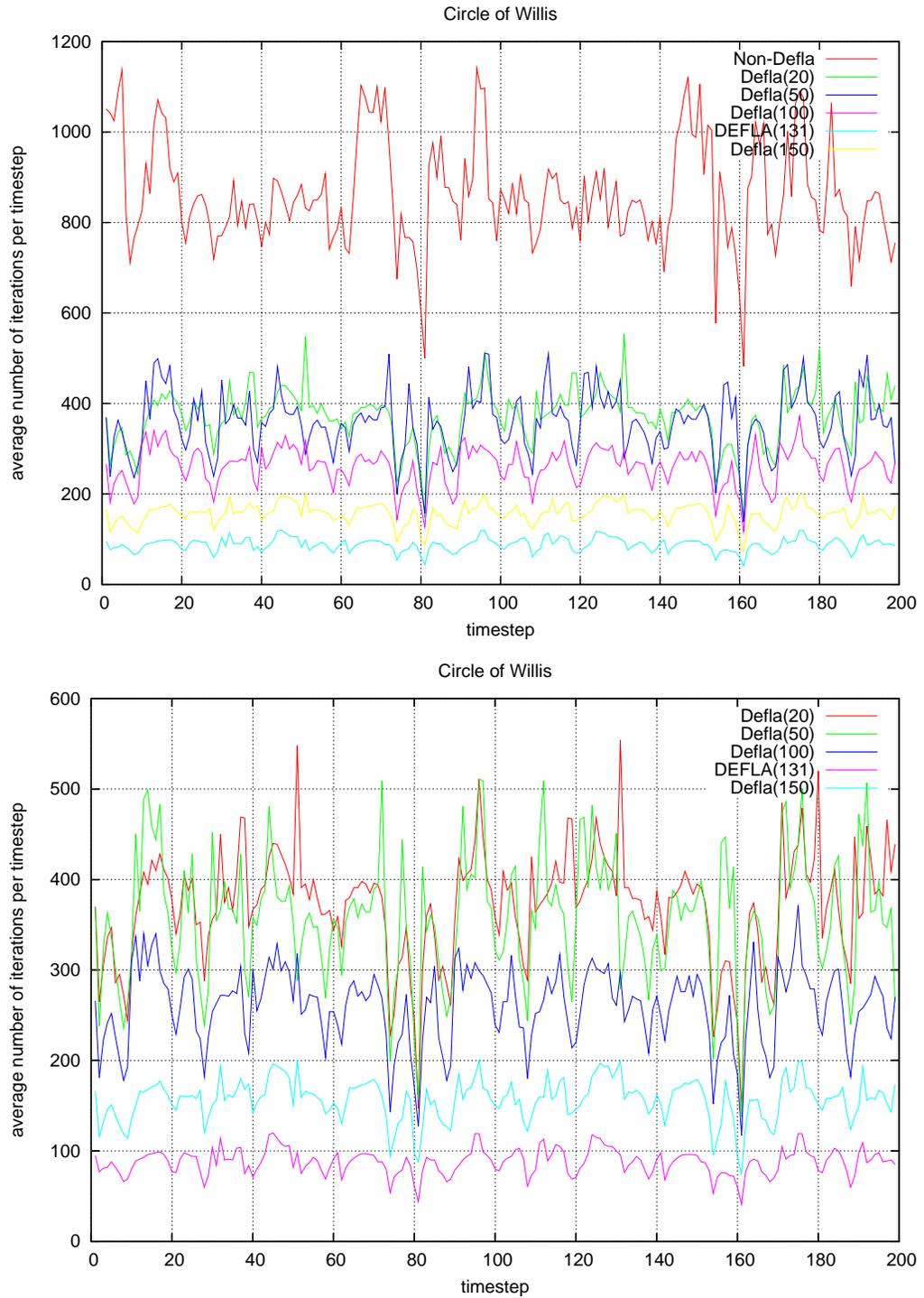


Figure 2.26: COW: Average number of iterations for the Pressure-Poisson solver at each timestep for all cases (top) and deflation cases only (bottom).

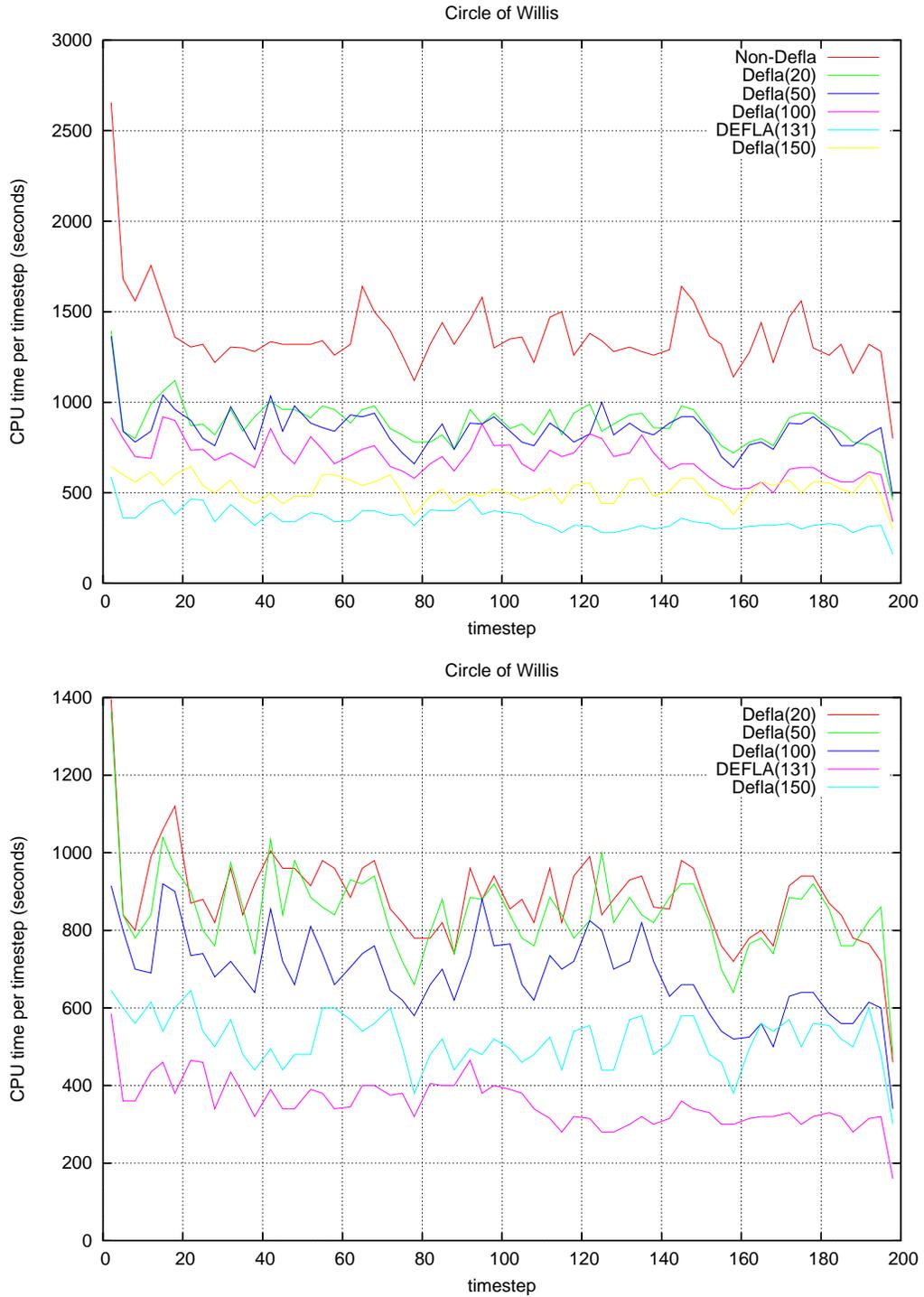


Figure 2.27: COW: CPU time at each timestep for all cases (top) and deflation cases only (bottom).

Circle of Willis - Timings

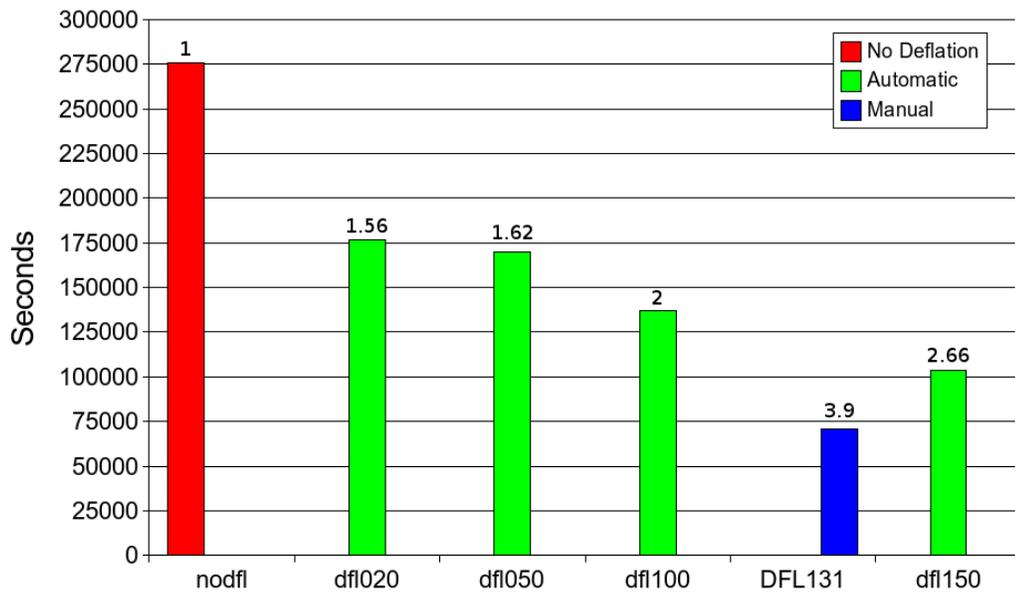


Figure 2.28: COW: Timings corresponding to a model of the Circle of Willis. The best speedup obtained was about 4.23 corresponding to 90 groups using the seedpoints alternative.

2.5.4 Cerebral Arteries

The last example is a patient-specific vascular model of the circle of Willis and the arterial network of the brain constructed from high resolution MRA images covering the entire brain of a normal subject[97]. The volume mesh had approximately 3.8M points and 19M elements. The groups were constructed using both methods. For the seedpoints alternative 512 and 978 equally spaced points were generated along the centerlines of the model. For the advancing layers method 250, 500, 1000, 2000 and 3500 groups were generated. Figures 2.29 shows the overall domain, as well as the deflation groups boundaries for the 1000 groups case (top) and the 978 groups case (bottom).

The pressure was prescribed (homogeneous bc) at the 3 inflows (bottom part), while a time-dependent velocity profile was prescribed at the 62 outflows. The flow waveforms were scaled with each outflow area in order to obtain the same wall shear stress at all the outflows (10 dyne/cm²).

Figure 2.30 depict the pressure drop (top) and wall shear stress distribution (bottom) at the peak systole of the second cardiac cycle. The differences in pressure observed between the left and right sides of the nodes is due to a bend in the vessel geometry which produces a significant pressure drop. Figure 2.31 shows a close-up view of this feature. This pressure drop also results in locally elevated wall shear stress in this region.

Figure 2.32 shows the average number of iterations required for the pressure Poisson solver at each timestep. The standard PCG solver has a mean value of 536 iterations per timestep, while for the deflated PCG solver this value ranges from 40 and 150 iterations per timestep. This implies that, in average, the number of iterations for the PCG solver have decreased between 3.5 and 13.5 times corresponding to the 250 and 3500 groups, respectively. As observed before, for the non-deflated case the standard PCG solver have not reached convergence as it reached the maximum pre-set limit of 1200 iterations in several timesteps.

Figure 2.33 depicts the CPU time required to complete each timestep. For the non-deflated case, the average CPU time is about 36m40s. For the deflated cases, the average

CPU time ranges between 10m25s and 16.45s seconds. The average speedup of each timestep is then located between 2.2 and 3.5 for the cases with 250 and 3500 groups, respectively. The bottom parts of figures 2.32 and 2.33 reveals that for all the cases as the number of groups increases, the number of iterations and average CPU time always decreases. Note that for this model there is no apparent distinction in the performance obtained between the methods used for generating the deflation subdomains. This is consistent with the fact that both methods generated similar group shapes as it can be seen in figure 2.29.

Figure 2.34 shows the timing for all the cases that were run. The best speedup obtained is about 3.8 corresponding to the 3500 groups case using the advancing layers method.

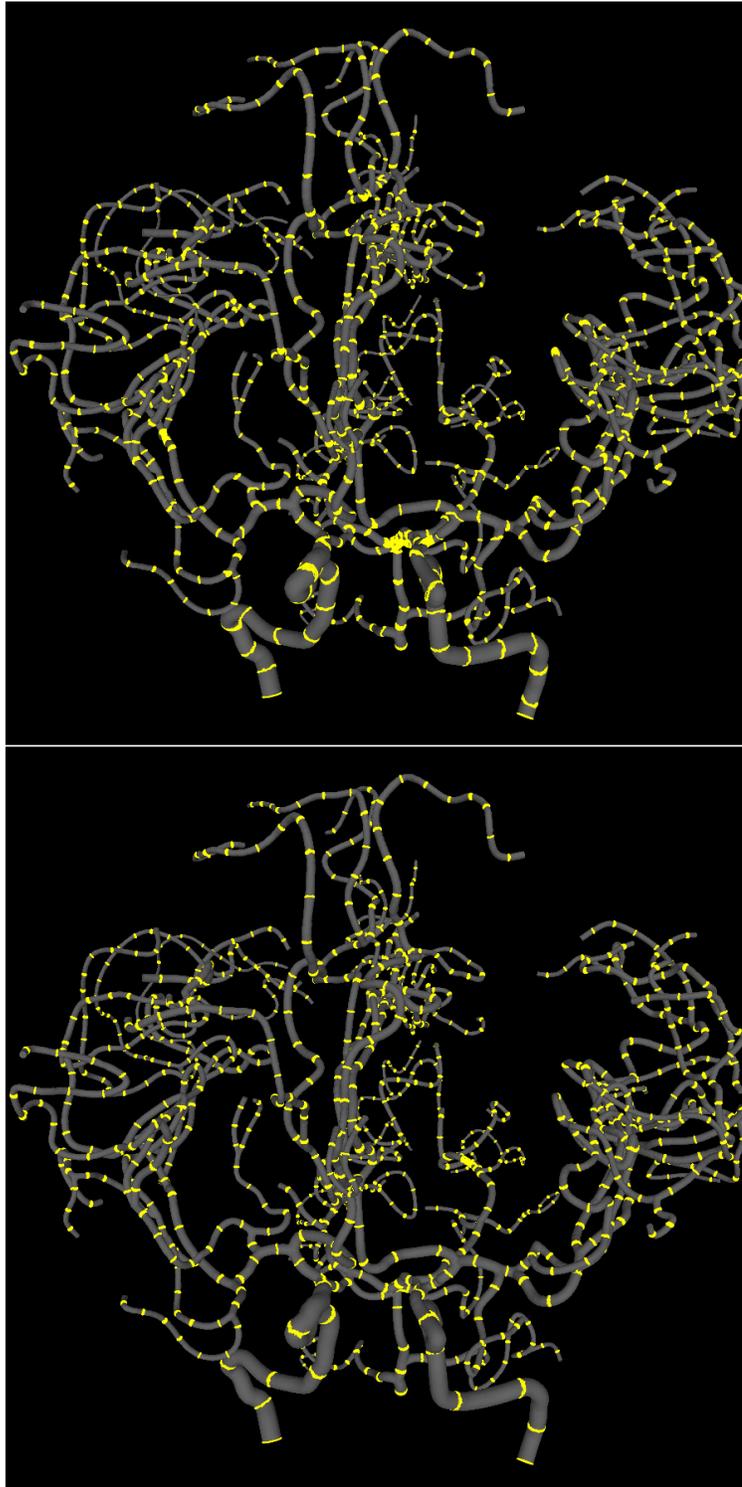


Figure 2.29: Cerebral Arteries: Deflation boundary subdomains corresponding to 1000 groups selected by the advancing layers method (top) and 978 groups selected by the seed-points alternative (bottom).

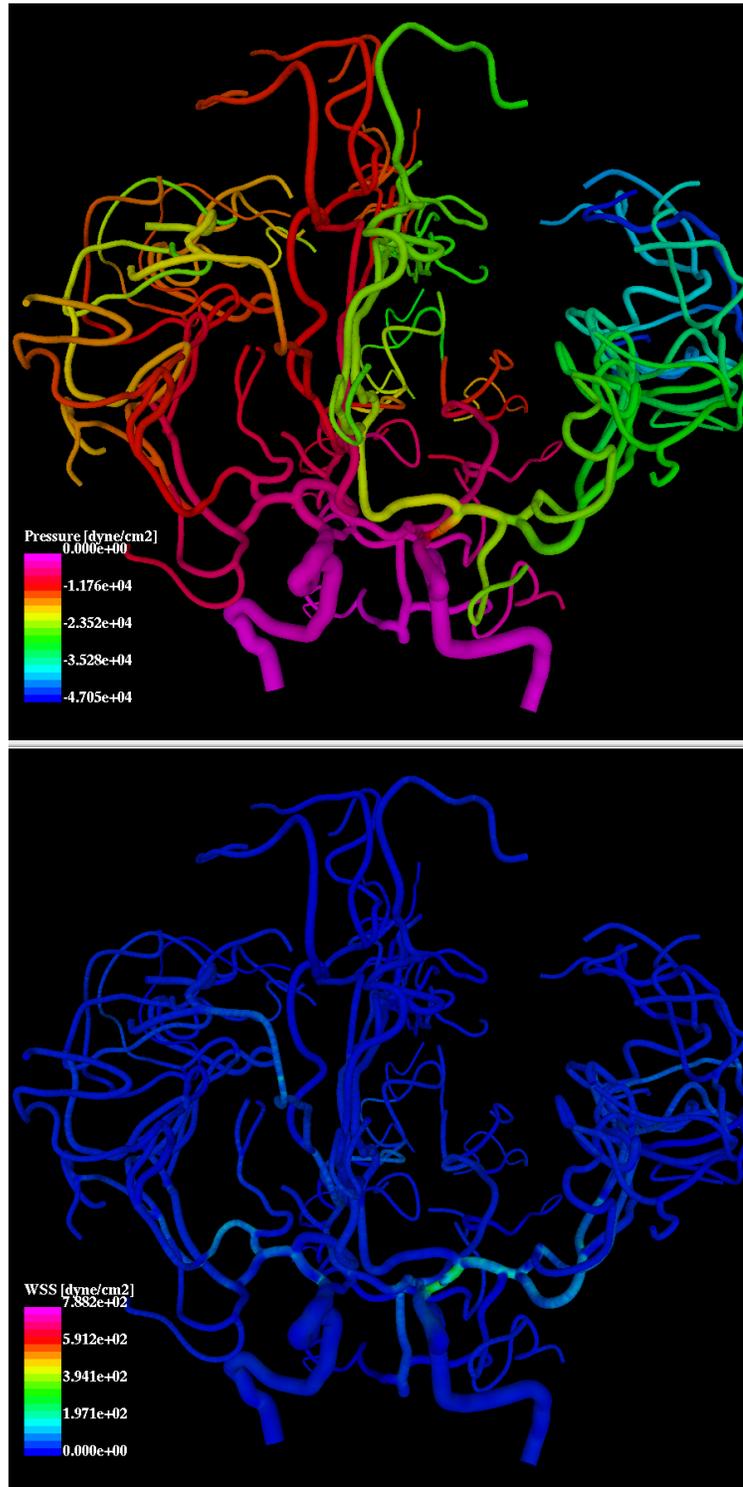


Figure 2.30: Cerebral Arteries: Pressure and Wall Shear Stress distribution at time $T = 1.4\text{sec}$.

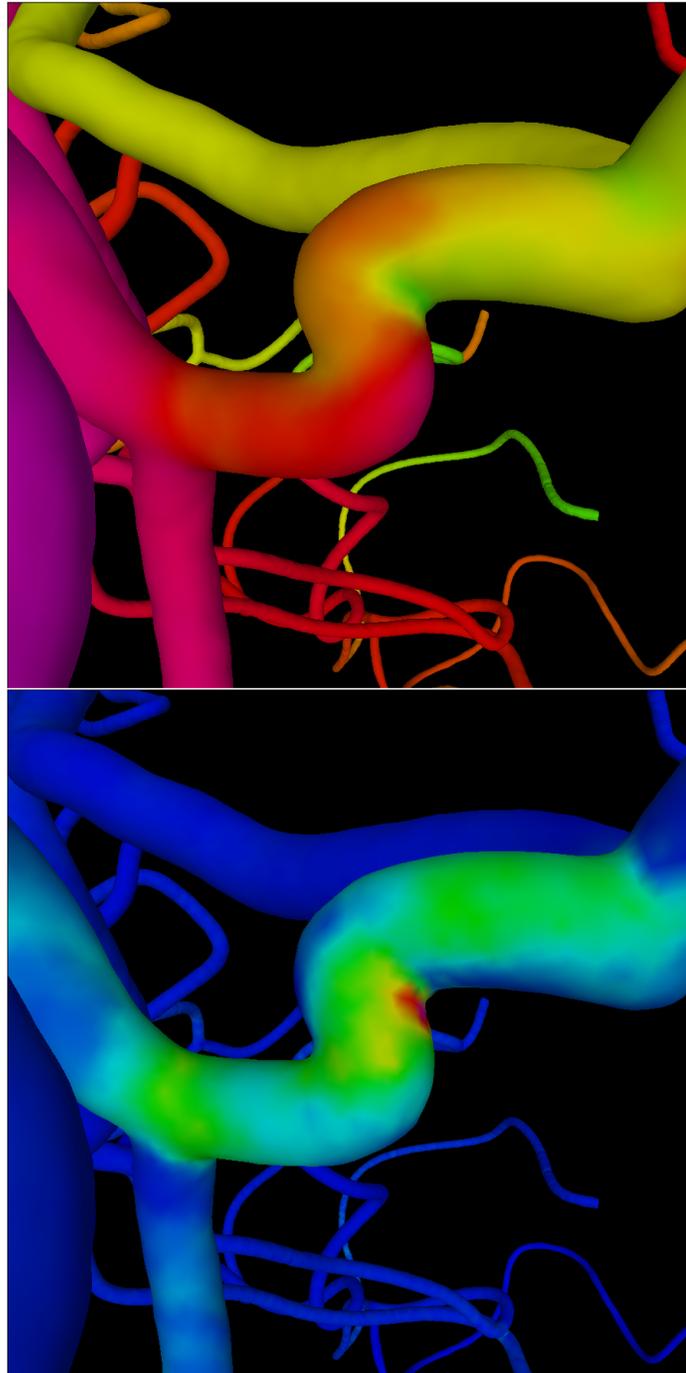


Figure 2.31: Cerebral Arteries: Huge pressure drop (top) at time $T = 1.4\text{sec.}$ due to an high bend in the vessel geometry. This pressure drop also produces a high wall shear stress region (bottom).

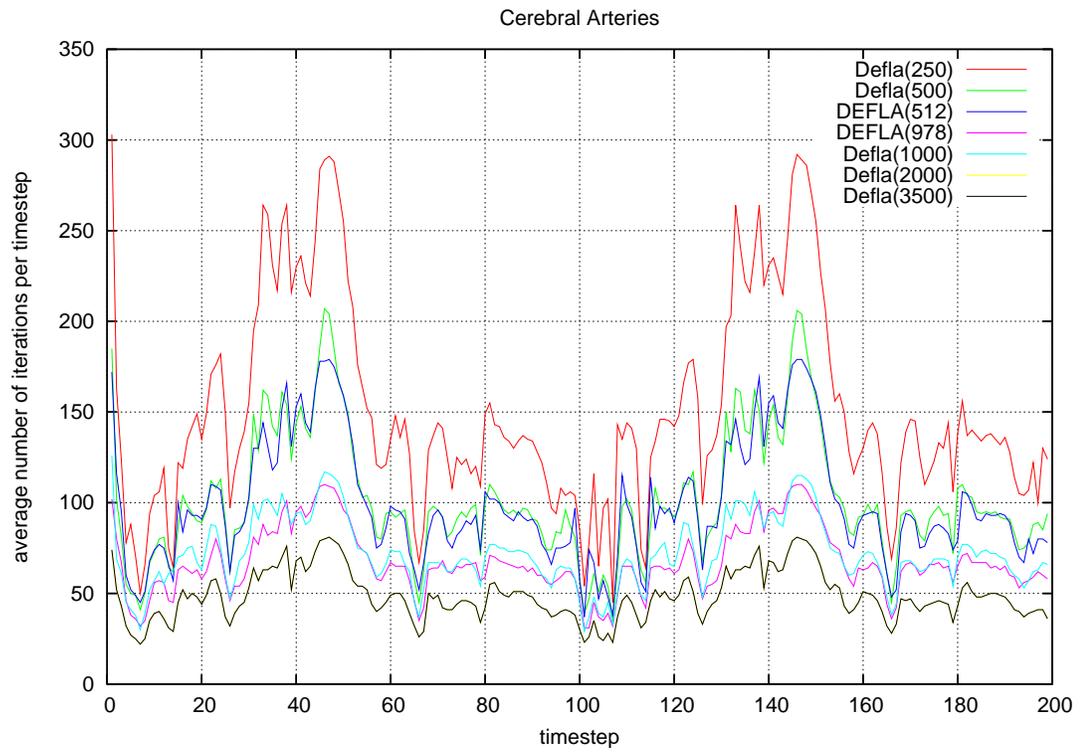
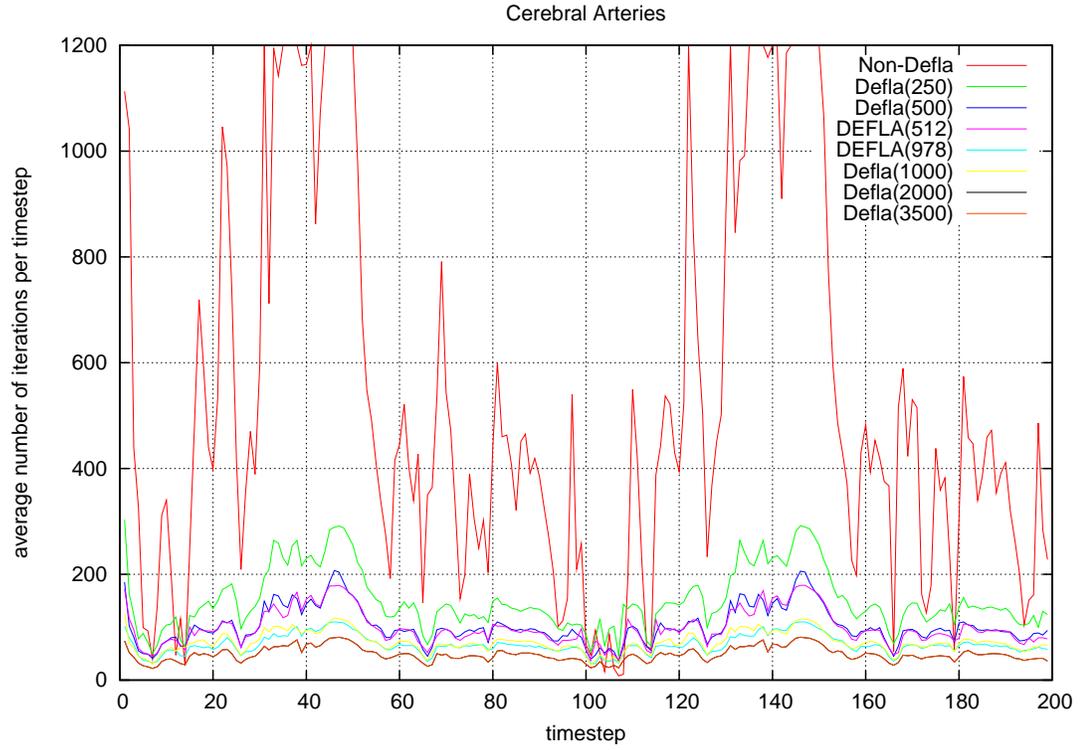


Figure 2.32: Cerebral Arteries: Average number of iterations for the pressure Poisson solver at each timestep for all the cases (top) and the deflation cases only (bottom).

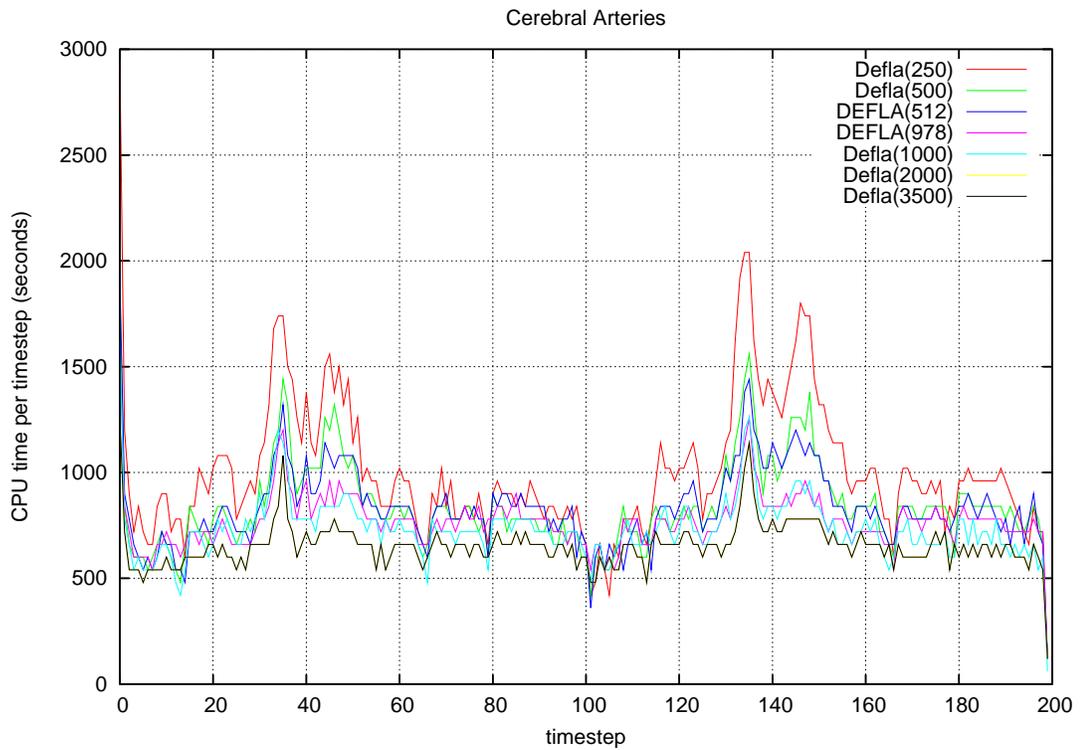
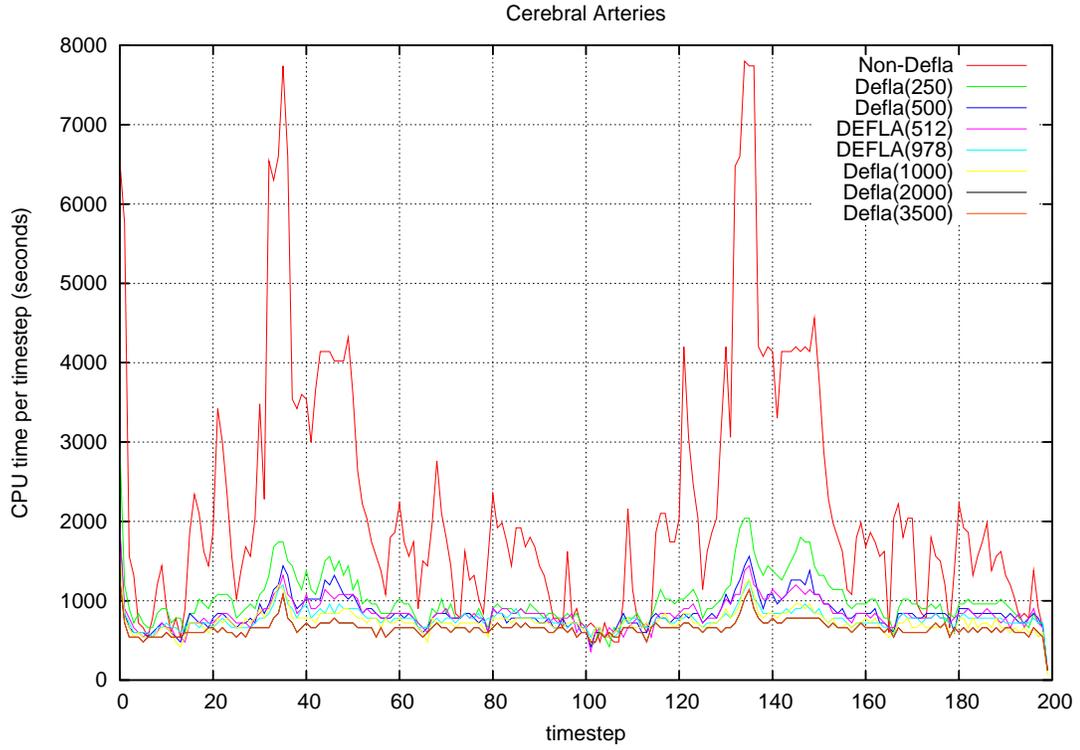


Figure 2.33: Cerebral Arteries: CPU time at each timestep corresponding to all the cases (top) and the deflation cases only (bottom).

Cerebral Arteries - Timings

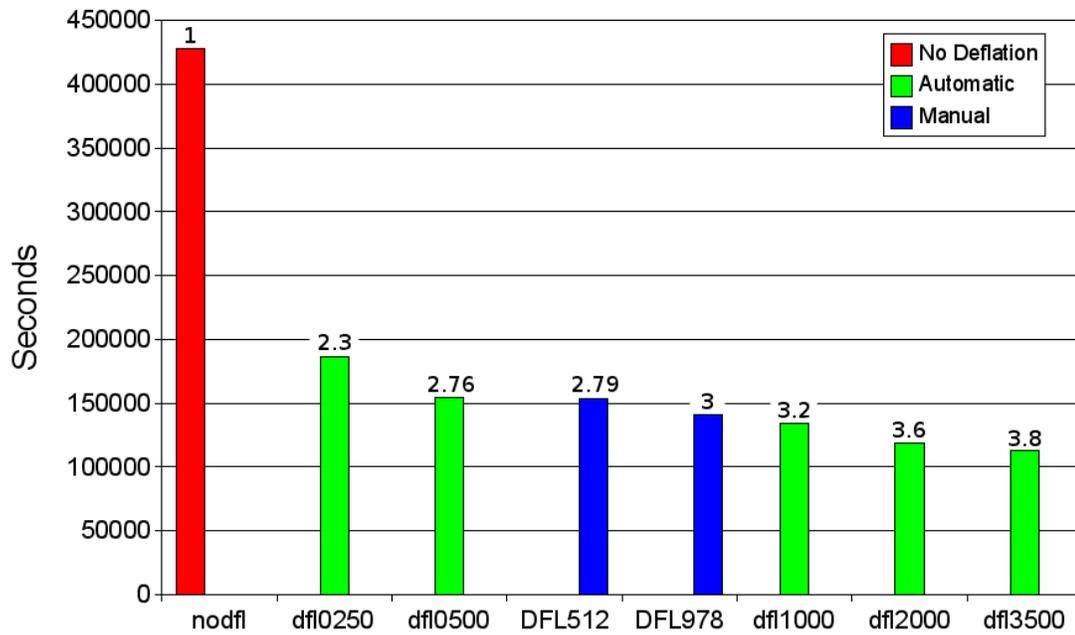


Figure 2.34: Cerebral Arteries: Timings corresponding to a model of the cerebral arteries. The best speedup obtained was about 3.8 corresponding to 3500 groups using the advancing layers method.

2.6 Summary

A Deflated Preconditioned Conjugate Gradients (DPCG) technique has been developed for the pressure Poisson equation within an incompressible flow solver. This technique was successfully tested on idealized cases. Several patient-specific image-based blood flow computations were also carried out to assess the performance of the deflated PCG solver on real cases ranging from relatively simple to quite complex geometries.

For all the cases, the solutions obtained with the deflated PCG solver coincided with those obtained with the standard PCG solver. The number of iteration was significantly reduced (up to a factor of 13) while speedups between 2 and 5 times were obtained in the solution of the incompressible flow equations.

Automatic (advancing layers) and manual (seedpoints) methods for constructing the deflation subdomains have been developed. These methods resulted in similar performances. However, in some cases, the manual method outperformed the automatic one, possibly because it produced more regular groups with more uniform pressure distributions.

In conclusion, this methodology has improved substantially the performance of the incompressible flow solver, making larger and more complex models practical. Although the methodology was developed in the context of blood flow simulations, it is general and can be used for any incompressible flow calculation.

Chapter 3: Virtual Stenting

3.1 Introduction

Currently there is an increased interest in using endovascular devices as flow diverters for the treatment of cerebral aneurysms. The main goal of these endovascular devices, such as stents, is to deviate the blood flow away from the aneurysm and promote aneurysm thrombosis and prevent its rupture. However, several studies have shown[98] that the specific stent design and its positioning may play an important role in how the blood flow is diverted. For this reason, a computational model to study the changes in the flow patterns before and after stenting is a highly valuable tool.

The computational modeling of patient-specific stenting is divided into two major steps: 1) Virtual Stenting: the process of virtually positioning a given stent within the anatomical model, and 2) Blood flow Modeling: the process of finding a solution to the governing equations for the blood flow in the presence of the stent. The objective of these calculations is to assess the efficacy of different stent designs or treatment plans in diverting the flow and to create a hemodynamic condition within the aneurysm favorable for thrombosis and aneurysm occlusion.

In what follows, a description of the methodology developed for the virtual deployment of stents within a given anatomical model is presented. The main aim of this tool is to provide with a realistic patient-specific anatomical model *after* stenting in such a way that it can be used for a numerical simulation of the blood flow. The final objective is to characterize the blood flow changes before and after stenting in order to assess the effectiveness of a given stent design on a patient-specific basis.

3.2 Methodology

The Virtual Stent Deployment methodology can be divided in four main steps: 1) parent vessel's centerline extraction, 2) initial cylindrical host surface generation, 3) host surface adaptation to the vessel wall, and 4) stent design mapping. In what follows a detailed description of each step is provided.

The first step is the extraction of the skeleton corresponding to the parent vessel. This is done by a minimal cost path construction algorithm[99]. The basic idea is first to compute the distance to the wall map within the anatomical model. Then the two endpoints of the skeleton (manually selected) are connected by the minimum arclength path that travels along the local maxima of the distance map. The output of this step is a collection of single connected points (edges) that represents the centerline of the parent vessel.

The second step consists in generating a cylindrical surface along the previously computed centerline. This is done by first computing an arclength parametrized cubic spline interpolation of the centerline points. Two arclength parameters s_0 and s_1 are then selected in such a way that the interpolated centerline between them satisfies the following: 1) it has the arclength of the targeted stent, and 2) it is in the required deployment position. These parameters are manually selected by trial and error. Finally a triangulated cylindrical surface is generated along the interpolated centerline between these values, where the radius of the cylinder is set to be the distance to the wall at each step. The output of this step is a triangulated cylindrical surface that lies completely inside the anatomical model, which roughly matches the vessel geometry and is already in the targeted deployment position.

The third step attempts to improve the fitting quality of the cylindrical surface to the vessel wall. This is carried out by a combination of external forces and internal smoothing forces applied to the triangulated surface. The external forces consist of an inflating force (or radial force) that is computed as the distance vector between the points in the triangulated surface and the centerline. This force has the effect of inflating the cylindrical surface while maintaining the cylindrical shape. The internal smoothing forces are based on the classical

smoothing Laplacian operator. These are basically attractive forces between each point in the triangulated surface and its first neighbors, i.e., all the points that are connected to it. The effect of this force is to keep the cylindrical surface smooth while it is deformed.

The vertices of the cylindrical surface are updated using a classical Newtonian law of motion resulting in the following set of equations:

$$m_i \frac{\partial^2 \mathbf{p}_i}{\partial t^2} + \gamma \frac{\partial \mathbf{p}_i}{\partial t} - \tilde{\alpha}_i f_{int}(\mathbf{p}_i) = \tilde{\beta}_i f_{ext}(\mathbf{p}_i) \quad (3.1)$$

where m_i is the i th vertex mass and γ is a damping parameter. The parameters $\tilde{\alpha}_i$ and $\tilde{\beta}_i$ controls the influence of the internal and external forces. The discretization of equation 3.1 using a fully explicit discretization scheme leads to

$$\mathbf{p}_i^{n+1} = \mathbf{p}_i^n + (1 - \gamma)(\mathbf{p}_i^n - \mathbf{p}_i^{n-1}) + \alpha_i f_{int}(\mathbf{p}_i^n) + \beta_i f_{ext}(\mathbf{p}_i^n) \quad (3.2)$$

where α_i and β_i are force weights including the point mass and the timestep. The stability of this scheme is guaranteed if α_i , β_i and γ lie inside $[0, 1/2]$, $[0, 1]$ and $[0, 1]$ respectively. For all the cases the parameters were set to be $\gamma = 1$, $\alpha_i = 0.01$ and $\beta_i = 0.0005$. These values were selected on a trial and error basis using several test cases.

Boundary conditions are applied to the points lying on the top and bottom ends of the cylinder. For these points only the components of the total force that are in the planes defined by the top and bottom ends of the cylinder are kept. This is done in order to maintain the original stent length and deployment position along the vessel.

The deformation process is performed interactively and stopped when most of the points of the cylindrical surface are on the vessel wall. This step has to be done interactively since there is no mechanism to prevent the cylindrical surface to enter into the aneurysm. When a point crosses the vessel wall, it is projected back onto the wall and kept fixed in that position for the rest of the simulation. Figure (3.1) depicts the deformed cylindrical surface, the vessel wall and the arrows representing the internal and external forces.

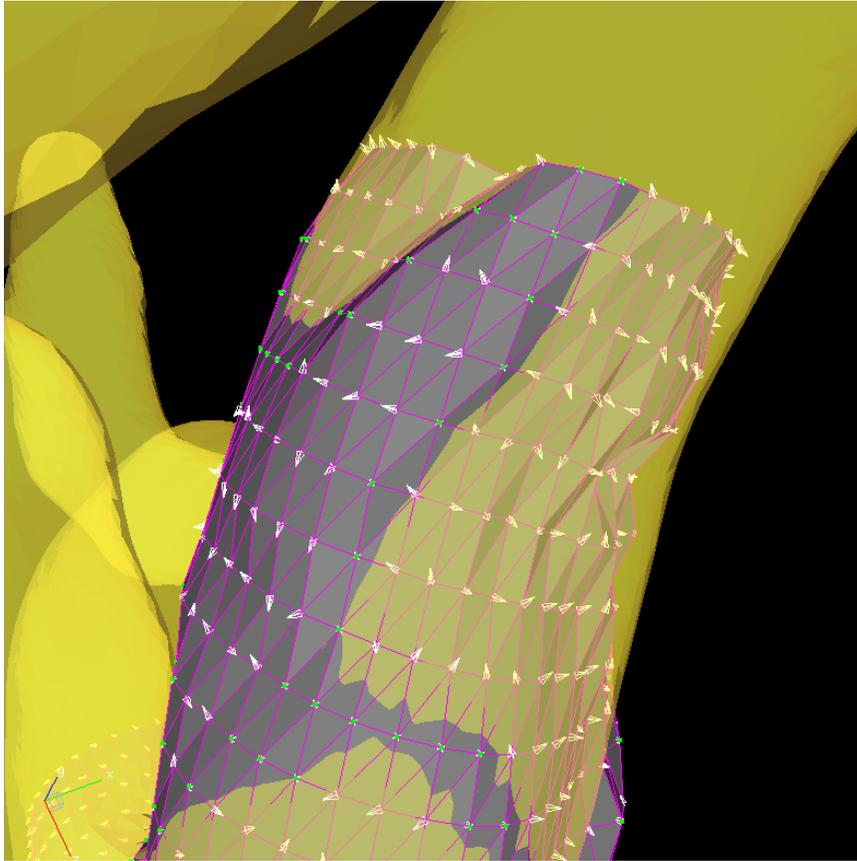


Figure 3.1: Virtual Stent Deployment: Intermediate stage during the deformation of a cylindrical surface in order to match the given vessel wall. The arrows represents the actual internal and external forces exerted over the cylinder under deformation.

The final step consists in using the deformed cylindrical surface as a host surface to map different stent designs. The stent designs are drawn on a 2D rectangle as a collection of connected lines with appropriate thickness. This 2D rectangle is a discrete surface that matches the host cylinder in the number and distribution of triangular elements. Using a simple coordinates transformation, the stent designs are mapped onto the deformed cylinder in order to obtain the final stent models in the deployed state. A constant phase angle can be provided at this step in order to rotate the deployed stent around the vessel.

The procedure to deploy a stent into a patient-specific vascular model of a cerebral aneurysm is illustrated in figure (3.2).

The blood flow simulation after stenting is computed using the finite element method

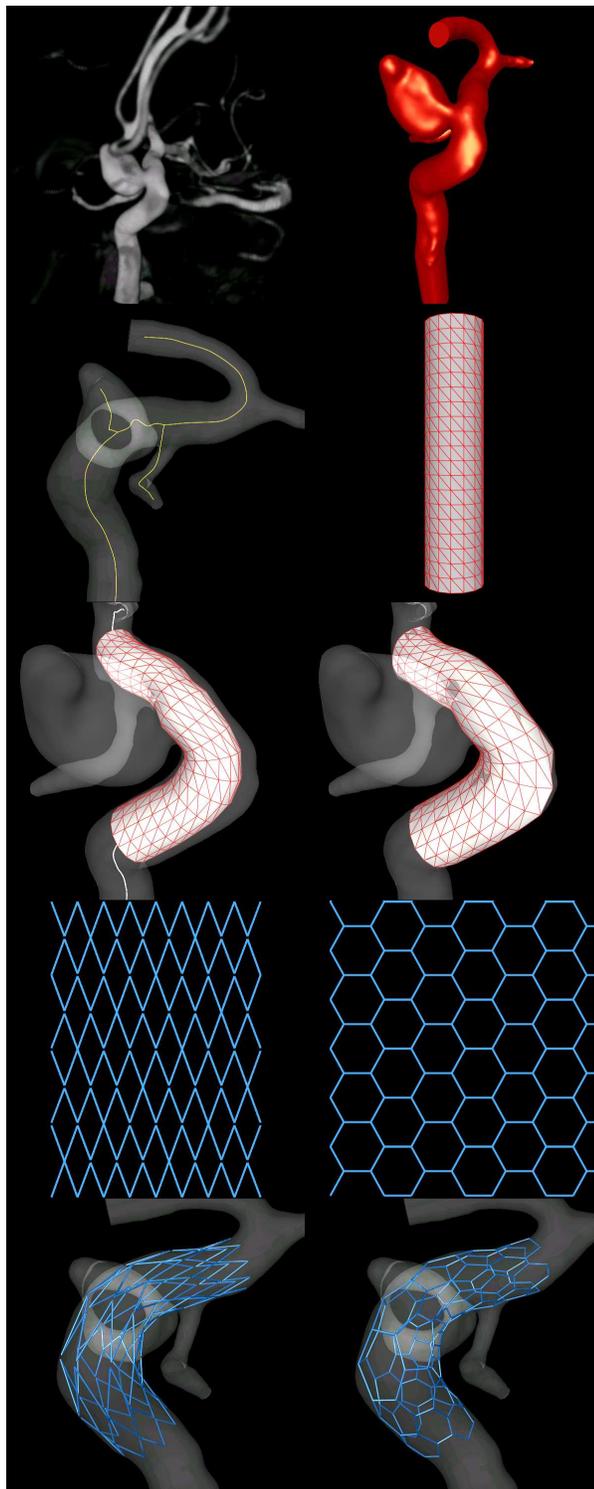


Figure 3.2: Virtual Stent Deployment: Different stages of the Virtual Stent Deployment methodology.

in conjunction with an adaptative embedded technique for unstructured grids[98]. This method greatly simplifies the flow simulation since it doesn't require a remeshing of the model after stenting. This methodology has been successfully applied in the simulation and characterization of several stents designs[98, 100, 101].

3.3 Evaluation

The Virtual Stent Deployment methodology was evaluated using an idealized glass model of a cerebral aneurysms with a Neuroform stent in place. Figure 3.3 shows a picture of the deployed stent within the phantom model.

A geometrical model of this idealized case was reconstructed from a 3-D rotational angiographic scan of the glass model. A Neuroform stent (Boston Scientific, Inc.) design was then virtually deployed using the methodology previously presented. Figure 3.4 shows the virtually deployed Neuroform stent thus obtained.

It can be seen from comparison between figure 3.3 and figure 3.4 that the stent was successfully positioned along and around the parent vessel. Also the length of the stent and its conformability to the vessel wall are well approximated. However the size of the gaps between the stent wires at the top of the model are not well approximated by the computational model. Note also how the wires of the stent in the bottom part of the phantom model are crossed with each other, while in the computational model this feature is not reproduced. These two discrepancies are due to the fact that the mapping step between the 2-D design and the deployed cylinder is a continuous mapping. Since the deployment methodology is only geometric, i.e. it does not include a mechanical modeling of the stent wires, this behavior is accentuated in regions of high curvature of the vessel wall and in places where the stent wires are not connected to each other.

However, these limitations are less important for stent designs with small closed cells and vessels with small curvature and it does not seem to affect substantially the overall solution of the blood flow field after stenting[98]. In what follows several examples of patient-specific models of stented cerebral aneurysms using this methodology are presented.

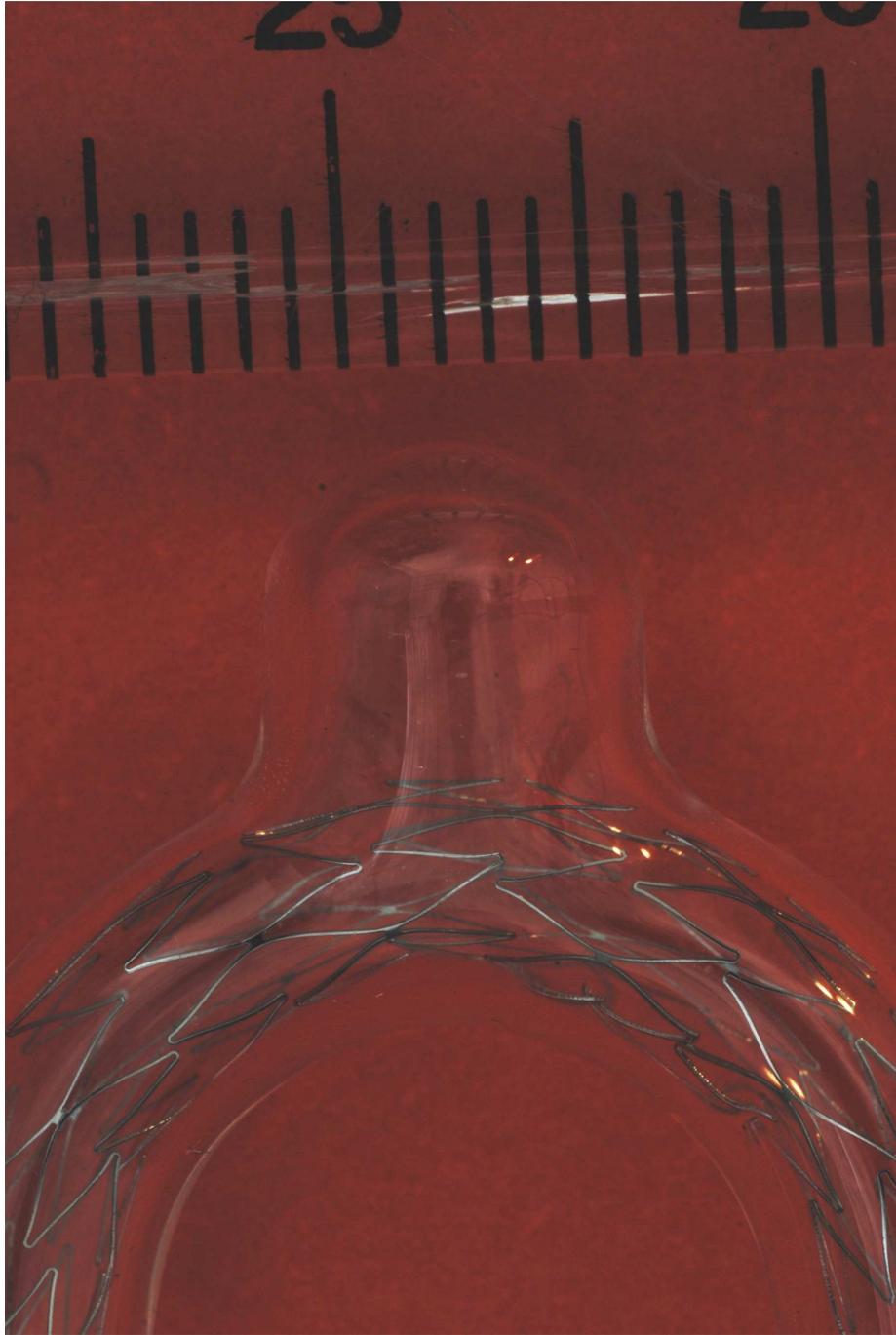


Figure 3.3: Phantom Case: Deployed stent (Neuroform) within an idealized model of an stent with its parent vessel made of glass.

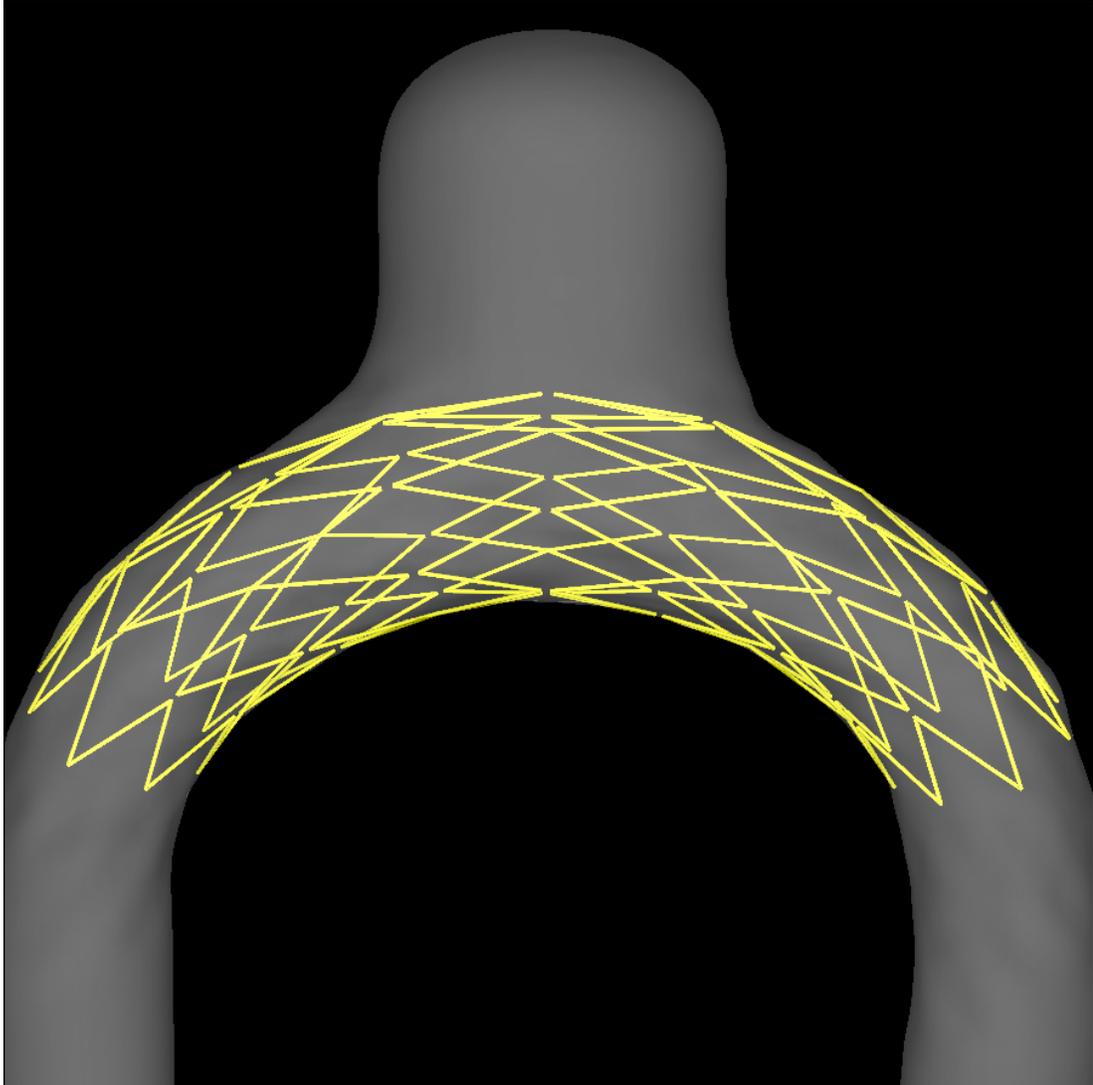


Figure 3.4: Phantom Case: Virtually Deployed stent using the Virtual Stent Deployment methodology.

3.4 Applications

The virtual stenting technique was demonstrated in several patient-specific cases of cerebral aneurysms. For all the cases, the vascular models were constructed from 3DRA images obtained during intravascular injection of a contrast agent.

In the first example, three different stent designs were tested in four patient-specific anatomical models. The goal of these simulations was to characterize the effectiveness of the different designs in the diversion of the blood flow.

The second example is a cerebral aneurysm for which clinical studies before and after stenting treatment are available. This provided the opportunity to compare the changes in the blood flow before and after stenting between the measured data and the computational model.

3.4.1 Comparison of Stent Designs

In this example, the virtual stenting technique was applied[100], in conjunction with an embedded technique, to compute the blood flow in four different patient-specific models of cerebral aneurysms with three stents of different design. Figure 3.5 shows the 3DRA images used to construct the vascular models and the reconstructed anatomical models with the various stent designs used. The first stent (third row) is made up of rhomboidal cells similar to the Neuroform stent (Boston Scientific, Inc.) while the remaining two stents have a helical-type design with wires oriented clockwise (fourth row) and counter-clockwise (last row). Simulations of the blood flow were carried out with and without stents. Physiologic pulsatile flow rates derived from measurements in normal subjects were prescribed at the inflow[51], while traction-free boundary conditions were imposed at the outflows. The same boundary conditions were used before and after stenting.

Figure 3.6 shows the streamlines at peak systole before and after stenting for all models and stents. For the first case (left column), when compared to the flow without stent, the Neuroform stent design (second row) diverts the inflow jet towards the center of the

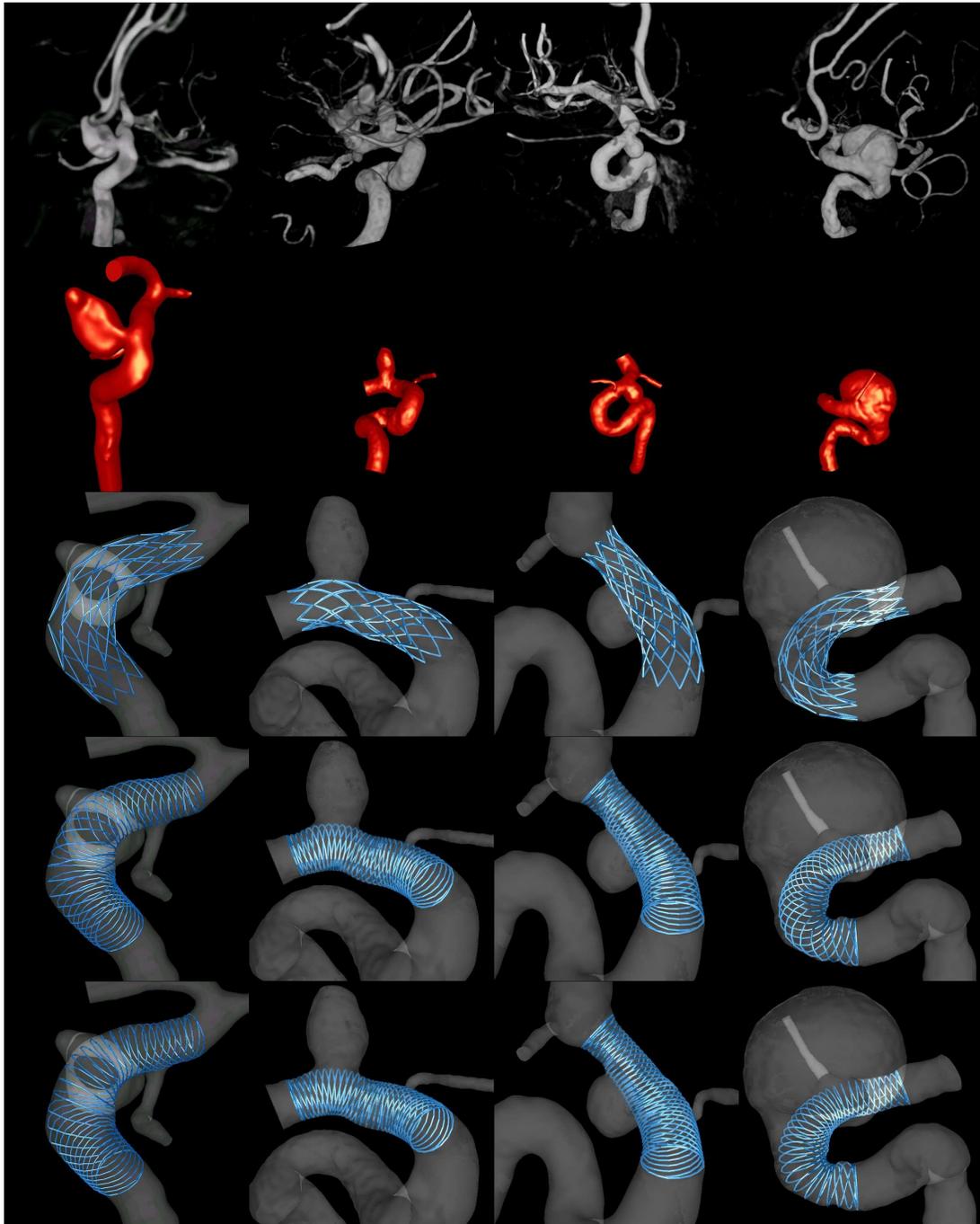


Figure 3.5: Comparison of Stent Designs: Original 3DRA image, computational model and three stents deployed into the model.

aneurysms dome. The helical stents divert the flow to the center of the aneurysms and prove even more effective compared to the Neuroform in reducing the inflow jet.

In the second case (second column), the Neuroform stent only reduces slightly the inflow jet in comparison to the pre-stented case. Similarly to the first case, the helical stents obstruct the flow into the aneurysms more effectively. In the pre-stented case the inflow into the aneurysm was at the distal part of the neck and the outflow was at the proximal part. This was not altered by the Neuroform stent. However, the helical stents shifted the location of the inflow and outflow to the sides of the neck. From this case it is clear the the inflow and outflow in the aneurysm depend on factors such as the stent used, the location of the aneurysm on the parent vessel, the shape of the aneurysm and the parent vessel and the type of inflow.

The third case (third column) shows that the inflow after stenting was closer to the proximal part of the aneurysm neck and the outflow was at the distal part.

Hemodynamics in the pre and post-stented cases for the fourth aneurysm (right column) was different due to the size, shape and location of the aneurysm on the parent vessel. In the pre-stented case the inflow was at the proximal part and the jet was impacting on the body of the aneurysm. The Neuroform slightly reduced the inflow, but increased the concentration of the inflow jet. As expected, the helical stents diffused the inflow jet to a larger extent with lower flow into the aneurysm.

These examples illustrate how the methodology can be used to understand the changes in the hemodynamics produced by different stents. This can be useful for selecting the best stent for a given aneurysm.

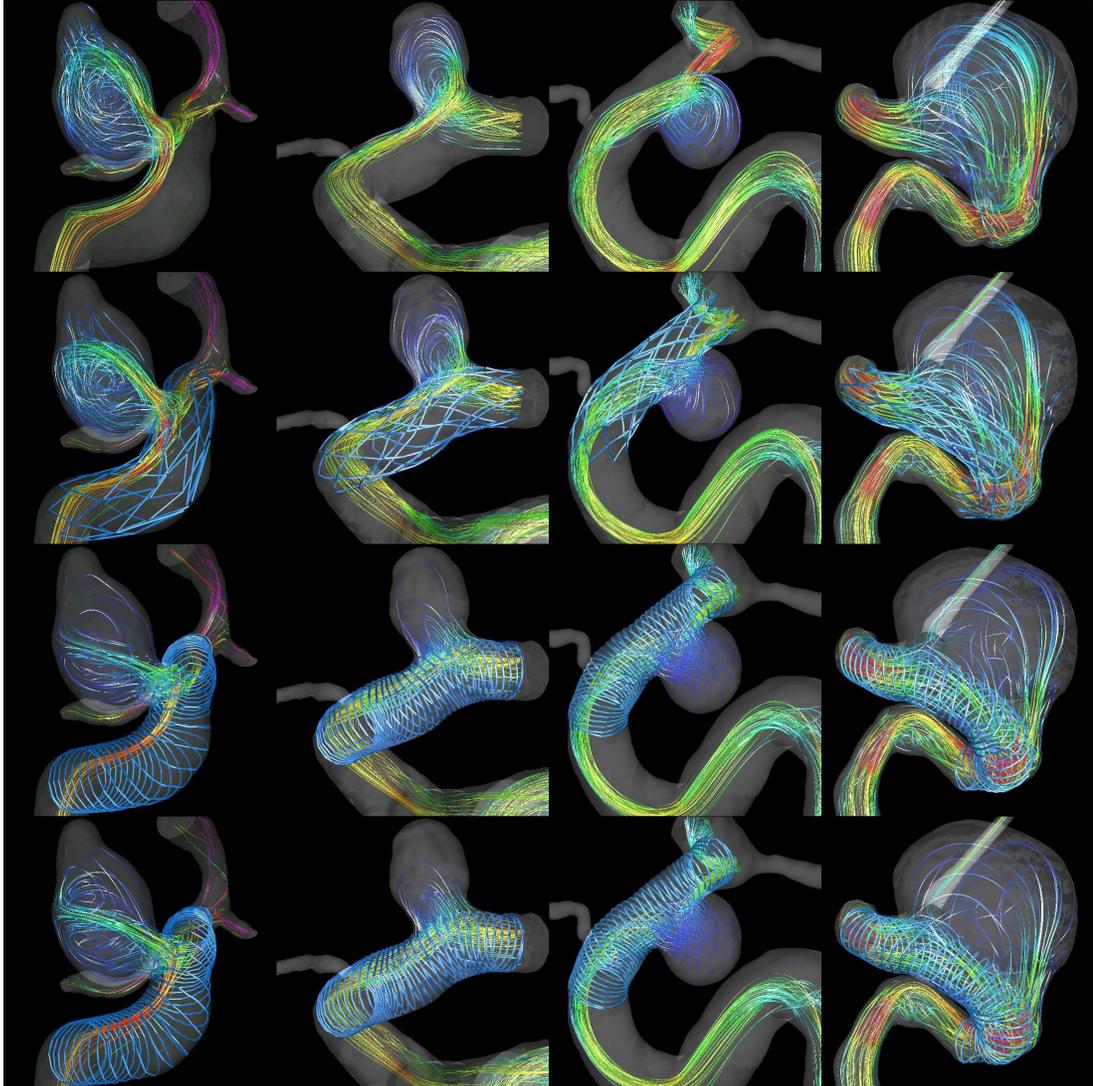


Figure 3.6: Comparison of Stent Designs: Streamlines for the various patient specific cases, pre-stented case (top row), with Neuroform stent (second row), left oriented helical stent (third row), and right oriented helical stent (bottom row).

3.4.2 Model of Aneurysm treated with stent

In this example, the alterations in the blood flow produced by a braided stent were analyzed. The case studied was a giant ($26mm$) intracranial aneurysm in the left internal carotid artery (ICA). The aneurysm was treated with a braided stent alone (Pipeline Neuroendovascular Device, Chestnut Medical Technologies, Inc). Rotational and conventional angiograms were acquired prior and immediately after stenting. A patient-specific model of the aneurysm and the parent artery before treatment was constructed from the 3D rotational angiogram. A model of the Pipeline stent was created following the geometry described in [102]

Figure 3.7 illustrates the modeling steps. This figure shows the rotational angiogram prior to stenting (top-left panel), the reconstructed vascular model (top-center panel), the deformed cylindrical support surface for stent deployment (top-right panel), the stent design (bottom-left panel), the deployed stent (bottom-center panel) and a detail of the mesh after embedding the stent and adaptive refinement (bottom-right panel).

The results of the numerical simulations of the hemodynamics before and after stenting the aneurysm are presented in Figure 3.8. This figure shows the intra-aneurysmal blood flow pattern at peak systole for the pre-stenting (left) and post-stenting (right) models. These visualizations show that before stenting there is a high speed inflow jet entering the aneurysm at the proximal end of the neck. This inflow jet produces a complex and unsteady flow pattern inside the aneurysm.

The placement of the stent produces a significant alteration of the blood flow pattern. In particular, the inflow jet is substantially diffused by the stent mesh resulting in a much simpler and more stable flow pattern with slower velocities and a smoother appearance. The peak flow rate entering the aneurysm was reduced from 2.32 ml/s to 1.47 ml/s (47% reduction).

Synthetic or 'virtual' angiograms were created to visualize the intra-aneurysmal flow characteristics. These virtual angiograms are visualizations of the passage of a simulated bolus of contrast transported by the blood flow. They are constructed by numerically

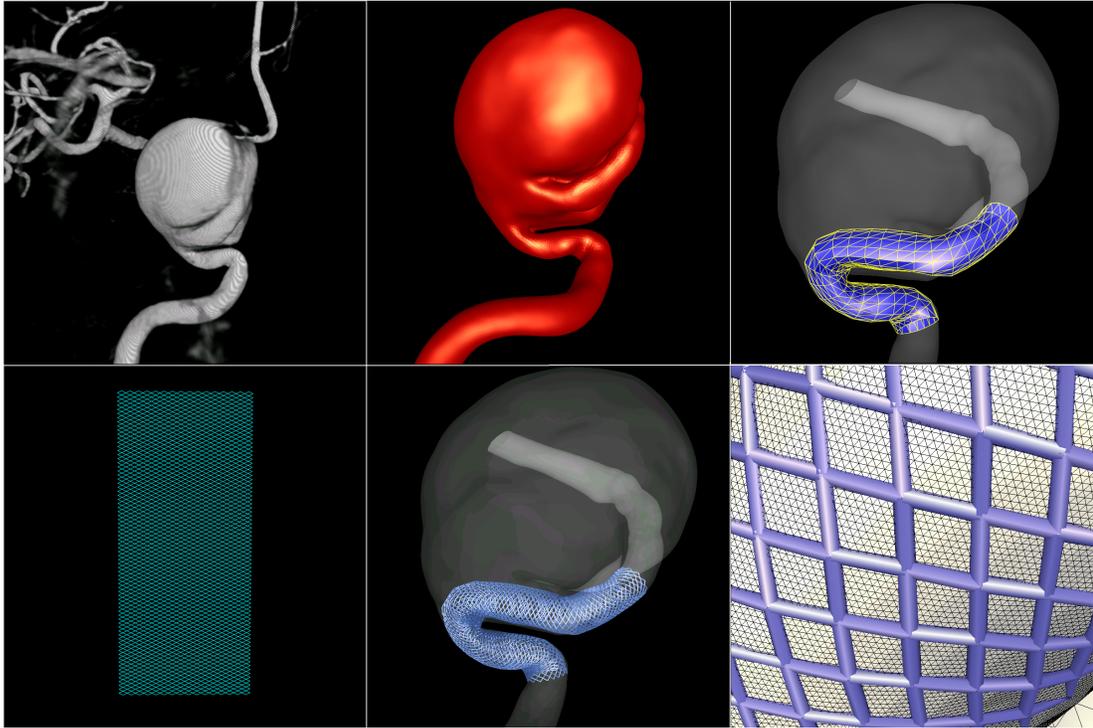


Figure 3.7: Pipeline: Top row: rotational angiography prior to treatment (left), vascular model (center), deformed cylinder (right). Bottom row: stent design (left), deployed stent (center), zoom up view of the refined mesh near the stent wires (right).

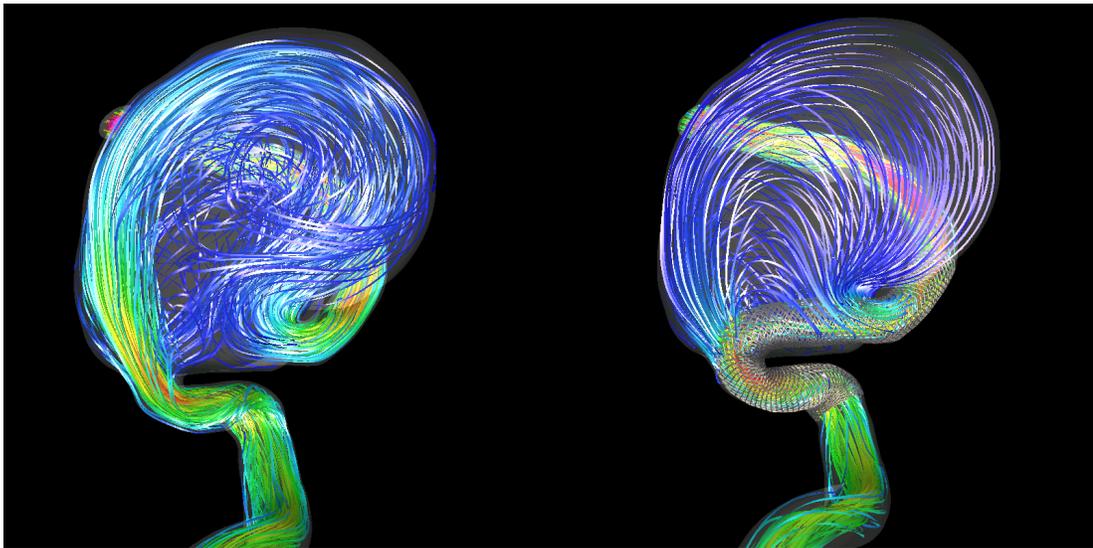


Figure 3.8: Pipeline: Flow pattern at peak systole before (left) and after (right) stenting.

solving the transport equation:

$$\phi_{,t} + \mathbf{v} \cdot \nabla \phi = 0 \quad (3.3)$$

where ϕ represents the dye concentration field, and \mathbf{v} is the velocity field obtained from the flow solver. This equation was numerically solved using a finite element scheme [103]. At the model inlet a time-dependent concentration with the shape of a Poisson distribution was imposed simulating four seconds injection of contrast. The concentration field was then advanced for 10 cardiac cycles assuming periodicity of the flow velocity field. The virtual angiograms were then created by a volume rendering technique using the mesh points with opacity and intensity modulated by the concentration field at each instant.

Selected frames of conventional and virtual angiograms are shown in Figure 3.9. The left panel of this figure shows four frames of the conventional (left) and virtual (right) angiograms before aneurysm stenting. The first three frames (from top to bottom) correspond to the filling phase, while the last frame corresponds to the washout phase. It can be seen that the main flow characteristics observed in the conventional angiogram are reasonably reproduced by the computational model. Namely, the location and size of the inflow jet and flow impaction zone, and the major vortical structures observed inside the aneurysm.

The right panel of Figure 3.9 shows four selected frames of the conventional (left) and virtual (right) right after stenting. These visualizations confirm the predicted alteration of the flow pattern from a complex to a simple flow type and slower flow velocities (i.e. increased residence time). They also show that little contrast reaches the distal part of the dome, while most of the dye concentrates near the inflow zone.

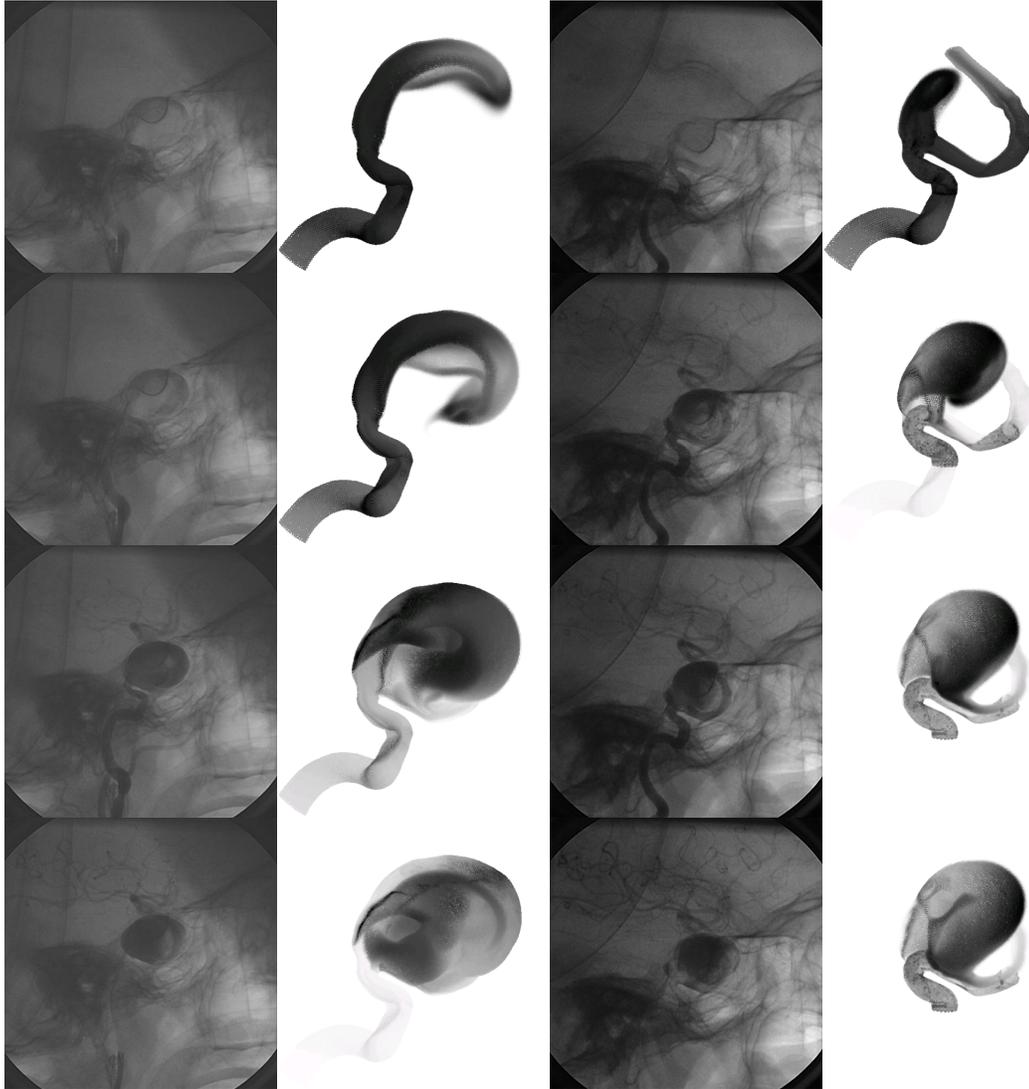


Figure 3.9: Pipeline: Conventional (gray background) and virtual (white background) angiogram prior to stenting (left panel) and right after stenting (right panel).

3.5 Summary

A novel methodology for virtual stent deployment in patient-specific models of cerebral aneurysms has been developed. This represents a first step towards an accurate and realistic method for modeling the hemodynamics of stents in aneurysms.

The technique was evaluated with a glass phantom of a stented aneurysm in a curved vessel. In addition, the methodology was demonstrated with several patient-specific models of cerebral aneurysms constructed from 3DRA images and different stents designs. One of the main advantages is that different stent designs can be easily interchanged allowing for a fast characterization of the blood flow patterns in order to determine the best option.

This methodology has several limitations: 1) it can produce unrealistic elongations of the stent wires in regions of high vessel curvature, specially for stents with large cells; 2) the gaps between unconnected wires could be different from the actual gaps; 3) it does not produce forshortening of the stent; and 4) it requires manual intervention to avoid leaking into the aneurysm. However, these limitations are less important for stent designs with small closed cells and vessels with small curvature.

Despite these limitations, the methodology was able to reproduce the flow alterations observed in a patient treated with a braided stent. Further improvements are needed in order to obtain a more accurate representation of the stent geometry in the deployed stage.

Chapter 4: Conclusions and Outlook

Two new methodologies that extended the capabilities of the computational modeling of cerebral aneurysms have been developed and tested, namely, 1) an acceleration technique for the incompressible Navier-Stokes solver and 2) a methodology to virtually deploy stents within anatomical models of cerebral aneurysms.

A Deflated Preconditioned Conjugate Gradients (DPCG) technique for the solution of the pressure Poisson equation within an inhouse incompressible flow solver (FEFLO) has been developed. The main idea is to use a coarser discretization of the domain to approximate the lowest eigenmodes of the system and therefore promote a faster convergence rate. This technique was developed in the context of computational modeling of cerebral aneurysms, but it is general and can be used for any incompressible flow problem. Several examples have shown a speedup between 3 and 5 in the total computational time compared to the standard PCG solver. The number of iterations has also been reduced by up to a factor of 12. This technique has successfully extended the capabilities of the flow solver and larger and more complex vascular models have become practical. Future work will explore the possibility of applying this deflation technique to non-symmetric cases with the GMRES method. This would allow to also reduce the computational time required to solve the advection terms of the Navier-Stokes equations.

A pioneer technique for the virtual deployment of stents within patient-specific models of cerebral aneurysms have been developed. This methodology was tested on several models of cerebral aneurysms constructed from 3DRA images and different stent designs. One of the main advantages of this technique is the possibility of easily interchange stent designs in order to characterize its performance for a given patient-specific model. However, this methodology has some limitations in dealing with stent designs with large cells and high curvature vessel walls. Despite this limitations, the methodology was able to reproduce the

flow alterations observed in a patient treated with a braided stent. Further improvements are needed in order to obtain a more accurate representation of the stent geometry in the deployed stent.

Bibliography

Bibliography

- [1] W. E. Stehbens, *Pathology of the Cerebral Blood Vessels*. C. V. Mosby, 1972, ch. Intracranial aneurysms, pp. 351–470.
- [2] B. Wier, “Unruptured intracranial aneurysms: a review,” *Journal of Neurosurgery*, vol. 96, pp. 3–42, 2002.
- [3] G. N. Foutrakis, H. Yonas, and R. J. Scwabassi, “Saccular aneurysms formation in curved and bifurcation arteries,” *American Journal of Neuroradiology*, vol. 20, pp. 1309–1317, 1999.
- [4] F. Tomasello, D. D’Avella, F. M. Salpietro, and M. Longo, “Asymptomatic aneurysms. literature meta-analysis and indications for treatment,” *Journal of Neurosurgical Sciences*, vol. 42, no. 1, pp. 47–51, 1998.
- [5] H. R. Winn, J. A. Jane, J. Taylor, D. Kaiser, and G. W. Britz, “Detection of asymptomatic incidental aneurysms: review of 4568 arteriograms,” *Journal of Neurosurgery*, vol. 96, no. 1, pp. 43–49, 2002.
- [6] F. H. Linn, G. J. Rinkel, A. Algra, and J. van Gijn, “Incidence of subarachnoid hemorrhage: role of region, year, and rate of computed tomography: a meta-analysis,” *Stroke*, vol. 27, no. 4, pp. 625–629, 1996.
- [7] M. Kaminogo, M. Yonekura, and S. Shibata, “Incidence and outcome of multiple intracranial aneurysms in a defined population,” *Stroke*, vol. 34, no. 1, pp. 16–21, 2003.
- [8] C. Kim, J. Cervos-Navarro, C. Patzold, Y. Tokuriki, Y. Takebe, and K. Hori, “In vivo study of flow pattern at human carotid bifurcation with regard to aneurysm development,” *Acta Neurochirurgica*, vol. 115, no. 3–4, pp. 112–117, 1992.
- [9] W. E. Stehbens, “Etiology of intracranial berry aneurysms,” *Journal of Neurosurgery*, vol. 70, pp. 823–831, 1989.
- [10] ———, “Histopathology of cerebral aneurysms,” *Archives of Neurology*, vol. 8, pp. 272–285, 1963.
- [11] K. N. T. Kayembe, M. Sasahara, and F. Hazama, “Cerebral aneurysms and variations of the circle of willis,” *Stroke*, vol. 15, pp. 846–850, 1984.
- [12] M. G. Yasargil, *Microneurosurgery Vol. 2*. Thieme, Stuttgart New York, 1984, pp. 75–170.

- [13] G. Salar and S. Mingrino, "Ligature of the cervical carotid artery for the treatment of intracranial carotid aneurysms: complications and late results," *Acta Neurochirurgica*, pp. 36–152, 1977.
- [14] M. Matsuda and et al., "Ruptured cerebral aneurysms associated with arterial occlusion," *Surgical Neurology*, vol. 20, no. 1, pp. 4–12, 1983.
- [15] G. Salar and S. Mingrino, "Development of intracranial saccular aneurysms: report of 2 cases," *Neurosurgery*, vol. 8, pp. 462–465, 1981.
- [16] G. Perret and H. Nishioka, "Report on the international study of intracranial aneurysms and subarachnoid hemorrhage. section vi, arteriovenous malformations: an analysis of 545 cases of craniocerebral arteriovenous malformations and fistulae reported on the cooperative study," *Journal of Neurosurgery*, vol. 25, pp. 467–490, 1966.
- [17] S. J. Peerless and C. G. Drake, *Management of aneurysms of the posterior circulation*. WB Saunders, New York, 1982, vol. 3, pp. 1715–1763.
- [18] H. J. Steiger, "Pathophysiology of development and rupture of cerebral aneurysms," *Acta Neurochirurgica*, pp. 1–57, 1990.
- [19] S. Kondo, "Cerebral aneurysms arising at non-branching sites. an experimental study," *Stroke*, vol. 28, pp. 398–403, 1997.
- [20] C. Kim and et al., "Establishment of experimental conditions for inducing saccular cerebral aneurysms in primates with special reference to hypertension," *Acta Neurochirurgica*, vol. 96, pp. 132–136, 1989.
- [21] S. Suzuki and et al., "Experimentally induced cerebral aneurysms in rats: Vi, hypertension," *Surgical Neurology*, vol. 52, pp. 494–500, 1980.
- [22] I. Negata and et al., "Experimentally induced cerebral aneurysms in rats: Vi, hypertension," *Surgical Neurology*, vol. 14, pp. 477–479, 1980.
- [23] N. Hashimoto and et al., "Experimentally induced cerebral aneurysms in rats: V, relation of hemodynamics in the circle of willis to formation of aneurysms." *Surgical Neurology*, vol. 13, pp. 41–45, 1980.
- [24] H. Nakatani, N. Hashimoto, and Kang, "Cerebral blood flow patterns at major vessel bifurcations and aneurysms in rats," *Journal of Neurosurgery*, vol. 74, pp. 258–262, 1991.
- [25] A. Hara, N. Yoshimi, and H. Mori, "Evidence for apoptosis in human intracranial aneurysms," *Neurological Research*, vol. 20, pp. 127–130, 1998.
- [26] R. J. Guzman, K. Abe, and C. Zarins, "Flow-induced arterial enlargement is inhibited by suppression of nitric oxide synthase activity in vivo," *Surgery*, vol. 122, pp. 273–279, 1997.

- [27] E. Sho and et al., “Blood flow decrease induces apoptosis of endothelial cells in previously dilated arteries resulting from chronic high blood flow,” *Arteriosclerosis, Thrombosis & Vascular Biology*, vol. 21, pp. 1139–1145, 2001.
- [28] S. Fukuda and et al., “Prevention of rat cerebral aneurysm formation by inhibition of nitric oxide synthase,” *Circulation*, vol. 101, no. 21, pp. 2532–2538, 2000.
- [29] T. M. Griffith, “Modulation of blood flow and tissue perfusion by endothelium-derived relaxing factor,” *Experimental Physiology*, vol. 779, pp. 873–913, 1994.
- [30] D. W. Liepsch, “Flow in tubes and arteries: a comparison,” *Biorheology*, vol. 23, pp. 395–433, 1986.
- [31] S. Moncada, R. M. J. Plamer, and E. A. Higgs, “Nitric oxide: physiology, pathology and pharmacology,” *Pharmacology Review*, vol. 43, pp. 109–142, 1991.
- [32] K. Moritake and et al., “Experimental studies on intracranial aneurysms (a preliminary report): Some biomechanical considerations on the wall structures of intracranial aneurysms and experimentally produced aneurysms,” *No shinkei Sheka*, vol. 1, pp. 115–123, 1973.
- [33] Y. P. Gobin, J. L. Counard, and P. Flaud, “*In vitro* study of haemodynamics in a giant saccular aneurysm model: influence of flow dynamics in the parent vessel and effects of coil embolization,” *Neuroradiology*, vol. 36, no. 7, pp. 530–536, 1994.
- [34] C. F. Gonzales, Y. I. Choi, and V. Ortega, “Intracranial aneurysms: flow analysis of their origin and progression,” *American Journal of Neuroradiology*, vol. 13, pp. 181–188, 1992.
- [35] A. C. Burleson, C. M. Strother, and V. T. Tiritto, “Computer modeling of intracranial saccular and lateral aneurysms for the study of their hemodynamics,” *Neurosurgery*, vol. 37, pp. 774–784, 1995.
- [36] H. Tenjin, F. Asakura, and Y. Nakahara, “Evaluation of intraaneurysmal blood velocity by time-density curve analysis and digital subtraction angiography,” *American Journal of Neuroradiology*, vol. 19, pp. 1303–1307, 1998.
- [37] H. Ujiie, H. Tachibana, and O. Hiramitsu, “Effects of size and shape (aspect ratio) on the hemodynamics of saccular aneurysms: a possible index for the surgical treatment of intracranial aneurysms,” *Neurosurgery*, vol. 45, pp. 119–130, 1999.
- [38] S. Tateshima, Y. Murayama, and J. P. Villablanca, “Intraaneurysmal flow dynamics study featuring an acrylic aneurysm model manufactured using computerized tomography angiogram as a mold,” *Journal of Neurosurgery*, vol. 95, no. 6, pp. 1020–1027, 2001.
- [39] T. Satoh, K. Onoda, and S. Tsuchimoto, “Visualization of intraaneurysmal flow patterns with transluminal flow images of 3d mr angiograms in conjunction with aneurysmal configurations,” *American Journal of Neuroradiology*, vol. 24, pp. 1436–1445, 2003.

- [40] T. M. Liou and S. N. Liou, “A review of *in vitro* studies of hemodynamics characteristics in terminal and lateral aneurysm models,” *National Scientific Council ROB(B)*, vol. 23, no. 4, pp. 133–148, 1999.
- [41] R. Bammer, T. A. Hope, M. Aksoy, and M. T. Alley, “Time-resolved 3d quantitative flow mri of the major intracranial vessels: initial experience and comparative evaluation at 1.5t and 3.0t in combination with parallel imaging,” *Magnetic Resonance in Medicine*, vol. 57, pp. 127–140, 2007.
- [42] S. Wetzel, S. Meckel, A. Frydrychowicz, L. Bonati, E. W. Radue, K. Scheffler, J. Hennig, and M. Markl, “In vivo assessment and visualization of intracranial arterial hemodynamics with flow-sensitized 4d mr imaging at 3t,” *American Journal of Neuroradiology*, vol. 28, pp. 433–438, 2007.
- [43] L. D. Jou, C. M. Quick, W. L. Young, M. T. Lawton, R. Higashida, A. Martin, and D. Saloner, “Computational approach to quantifying hemodynamic forces in giant cerebral aneurysms,” *American Journal of Neuroradiology*, vol. 24, no. 9, pp. 1804–1810, 2003.
- [44] D. A. Steinman, J. S. Milner, C. J. Norley, S. P. Lownie, and D. W. Holdsworth, “Image-based computational simulation of flow dynamics in a giant intracranial aneurysm,” *American Journal of Neuroradiology*, vol. 24, no. 4, pp. 559–566, 2003.
- [45] J. R. Cebal, M. Hernandez, A. F. Frangi, C. M. Putman, R. Pergolizzi, and J. E. Burgess, “Subject-specific modeling of intracranial aneurysms,” *SPIE Medical Imaging*, vol. 5369, pp. 319–327, 2004.
- [46] T. Hassan, M. Ezura, E. V. Timofeev, T. Tominaga, T. Saito, A. Takahashi, K. Takayama, and T. Yoshimoto, “Computational simulation of therapeutic parent artery occlusion to treat giant vertebrobasilar aneurysm,” *American Journal of Neuroradiology*, vol. 25, no. 1, pp. 63–68, 2004.
- [47] J. R. Cebal, M. A. Castro, R. Löhner, O. Soto, P. J. Yim, and N. Alperin, “Finite element modeling of the circle of willis from magnetic resonance data,” *Proc. SPIE Medical Imaging*, vol. 5031, pp. 11–21, 2003.
- [48] J. R. Cebal, M. A. Castro, J. E. Burgess, R. Pergolizzi, M. J. Sheridan, and C. M. Putman, “Characterization of cerebral aneurysm for assessing risk of rupture using patient-specific computational hemodynamics models,” *American Journal of Neuroradiology*, vol. 26, pp. 2550–2559, 2005.
- [49] C. A. Taylor and M. T. Draney, “Computational techniques in therapeutic decision-making,” *Computer Assisted Surgery*, vol. 4, pp. 231–247, 1999.
- [50] J. R. Cebal and R. Löhner, “Efficient simulation of blood flow past complex endovascular devices using an adaptative embedding technique,” *IEEE Transactions in Medical Imaging*, vol. 24, no. 4, pp. 468–477, 2005.
- [51] J. R. Cebal, M. A. Castro, S. Appanaboyina, C. M. Putman, D. Millán, and A. F. Frangi, “Efficient pipeline for image-based patient-specific analysis of cerebral

- aneurysm hemodynamics: Technique and sensitivity,” *IEEE Transactions in Medical Imaging*, vol. 24, no. 1, pp. 457–467, 2005.
- [52] J. R. Cebal and R. Lohner, “From medical images to anatomically accurate finite element grids.” *International Journal for Numerical Methods in Engineering*, vol. 51, pp. 985–1008, 2001.
- [53] J. Beutel and M. Sonka, *Handbook of medical imaging, Volume 2: Medical image processing and analysis*, ser. SPIE Press Monograph. SPIE-International Society for Optical Engine, 2000, vol. PM80.
- [54] J. Weickert, *Anisotropic diffusion in image processing*, ser. ECMI. Stuttgart, Germany: Teubner-Verlag, 1998.
- [55] A. Frangi, “Multiscale vessel enhancement filtering,” *Lecture Notes in Computer Science*, vol. 1496, pp. 130–137, 1998.
- [56] P. J. Yim, G. Boudewijn, B. Vasbinder, V. B. Ho, and P. L. Choyke, “Isosurfaces as deformable models for magnetic resonance angiography,” *IEEE Transactions on Medical Imaging*, vol. 22, no. 7, pp. 875–881, 2003.
- [57] R. Lohner, *Applied CFD techniques: An Introduction based on Finite Element Methods*, 2nd ed. Wiley, 2008.
- [58] ———, “Extensions and improvements of the advancing front grid generation technique,” *Computer Methods in Applied Mechanics and Engineering*, vol. 5, pp. 119–132, 1996.
- [59] J. Mazumdar, *Biofluid mechanics*. World Scientific, 1992.
- [60] P. K. Kundu and I. M. Cohen, *Fluid Mechanics*. Elsevier, 2004.
- [61] J. R. Cebal, M. A. Castro, O. Soto, R. Löhner, and N. Alperin, “Blood-flow models of the circle of willis from magnetic resonance data,” *J. Eng. Math.*, vol. 47, pp. 369–386, 2003.
- [62] J. R. Womersley, “Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known,” *Journal of Physiology*, vol. 127, pp. 553–563, 1955.
- [63] C. A. Taylor, T. J. R. Hughes, and C. K. Zarins, “Finite element modeling of blood flow in arteries,” *Computer Methods in Applied Mechanics and Engineering*, vol. 158, pp. 155–196, 1998.
- [64] J. R. Cebal, P. J. Yim, , R. Löhner, O. Soto, and P. L. Choyke, “Blood flow modeling in carotid arteries using computational fluid dynamics and magnetic resonance imaging,” *Acad. Radiol.*, vol. 9, pp. 1286–1299, 2002.
- [65] S. Z. Zhao, X. Y. Xu, A. D. Hughes, S. A. Thom, A. V. Stanton, and B. Ariff, “Blood flow and vessel mechanics in a physiologically realistic model of a human carotid arterial bifurcation,” *J. Biomech.*, vol. 33, pp. 975–984, 2000.

- [66] L. Dempere-Marco, E. Oubel, M. A. Castro, C. M. Putman, A. F. Frangi, and J. R. Cebal, “Estimation of wall motion in intracranial aneurysms and its effects on hemodynamics patterns,” *Lecture Notes in Computer Sciences*, vol. 4191, pp. 438–445, 2006.
- [67] E. Oubel, M. DeCraene, C. M. Putman, J. R. Cebal, and A. F. Frangi, “Analysis of intracranial aneurysm wall motion and its effects on hemodynamic patterns,” in *SPIE Medical Imaging*, 2007.
- [68] X. He and D. N. Ku, “Pulsatile flow in the human left coronary artery bifurcation: Average conditions,” *Journal of Biomechanical Engineering*, vol. 118, pp. 74–82, 1996.
- [69] J. R. Cebal, M. A. Castro, C. M. Putman, and N. Alperin, “Flow-area relationship in internal carotid and vertebral arteries,” *Physiol Meas*, vol. 29, pp. 585–594, 2008.
- [70] I. E. Vignon-Clementel, C. A. Figueroa, K. E. Jansen, and C. A. Taylor, “Outflow boundary conditions for three-dimensional finite element modeling of blood flow and pressure in arteries,” *Comput. Methods. Appl. Mech. Engrg*, vol. 195, pp. 3776–3786, 2006.
- [71] R. Aubry, F. Mut, R. Löhner, and J. R. Cebal, “Deflated preconditioned conjugate gradients solvers for the pressure poisson equation,” Submitted February 2008, journal of Computational Physics.
- [72] F. Thomaset, *Implementation of Finite Element Methods for Navier-Stokes Equations*. Springer-Verlag, 1981.
- [73] M. D. Gunsburger and R. Nicolaides, *Incompressible Computational Fluid Dynamics: Trends and Advances*. Cambridge University Press, 1993.
- [74] M. M. Hafez, *Numerical Simulation of Incompressible Flows*. World Scientific, 2003.
- [75] P. F. Fisher, “Projection techniques for iterative solution of $ax=b$ with successive right-hand sides,” *Comp. Meth. Appl. Mech. Eng.*, vol. 163, pp. 193–204, 1998.
- [76] R. Löhner, “Projective prediction of pressure increments,” *Comm. Num. Meth. Eng.*, vol. 21, no. 4, pp. 201–207, 2005.
- [77] D. Martin and R. Löhner, “An implicit linelet-based solver for incompressible flows,” in *AIAA-92-0668*, 1992.
- [78] O. Soto, R. Löhner, and F. Camelli, “A linelet preconditioner for incompressible flows,” *Int. J. Num. Meth. Heat and Fluid Flow*, vol. 13, no. 1, pp. 133–147, 2003.
- [79] R. Löhner, “Multistage explicit advective prediction for projection-type incompressible flow solvers,” *Journal of Computational Physics*, vol. 195, pp. 143–152, 2004.
- [80] R. Löhner, C. Yang, J. R. Cebal, F. Camelli, O. Soto, and J. Waltz, “Improving the speed and accuracy of projection-type incompressible flow solvers,” *Comp. Meth. Appl. Mech. Eng.*, vol. 195, no. 23–24, pp. 3087–3109, 2006.

- [81] B. Allesandrini and G. Delhommeau, “A multigrid velocity-pressure-free surface elevation fully coupled solver for calculation of turbulent incompressible flow around a hull,” in *Proceeding of the 21st Symposium on Naval Hydrodynamics*, Trondheim, Norway, June 1996.
- [82] J. B. Bell, P. Colella, and H. Glaz, “A second-order projection method for the navier-stokes equations,” *Journal of Computational Physics*, vol. 85, no. 2, pp. 257–283, 1989.
- [83] J. B. Bell and D. L. Marcus, “A second-order projection method for variable-density flows,” *Journal of Computational Physics*, vol. 101, no. 2, pp. 334–348, 1992.
- [84] N. H. Chen, “Multigrid methods for the incompressible navier-stokes problem on three-dimensional unstructured meshes,” Ph.D. dissertation, George Mason University, 2004.
- [85] J. Waltz and R. Löhner, “An implicit linelet-based solver for incompressible flows,” in *AIAA-00-0925*, 2000.
- [86] J. Waltz, “Unstructured multigrid methods,” Ph.D. dissertation, George Mason University, 2000.
- [87] M. R. Hestenes and E. Stiefel, “Method of conjugate gradients for solving linear systems,” *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.
- [88] Y. Saad, *Iterative methods for sparse linear systems*, 2nd ed. SIAM, 2003.
- [89] A. van der Sluis and H. A. van der Vorst, “The rate of convergence of conjugate gradients,” *Numerische Mathematik*, vol. 48, pp. 543–560, 1986.
- [90] R. A. Nicolaides, “Deflation of conjugate gradients with applications to boundary value problems,” *SIAM J. Numer. Anal.*, vol. 24, no. 2, pp. 355–365, 1987.
- [91] L. Mansfield, “On the use of deflation to improve the convergence of the conjugate gradient iteration,” *Comm. Appl. Num. Meth.*, vol. 4, pp. 151–156, 1988.
- [92] ———, “On the conjugate gradient solution of the schur complement system obtained from domain decomposition,” *SIAM J. Numer. Anal.*, vol. 27, pp. 1612–1620, 1990.
- [93] J. Erhel, K. Burrage, and B. Pohl, “Restarted gmres preconditioned by deflation,” *J. Comput. Appl. Math.*, vol. 69, pp. 303–318, 1996.
- [94] Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc’h, “A deflated version of the conjugate gradient algorithm,” *SIAM J. Sci Comput.*, vol. 21, no. 5, pp. 1909–1926, 2000.
- [95] R. Löhner, “Multistage explicit advective prediction for projection-type incompressible flow solvers,” *Journal of Computational Physics*, vol. 195, pp. 143–152, 2004.
- [96] M. A. Castro, C. M. Putman, and J. R. Cebal, “Computational fluid dynamics modeling of intracranial aneurysms: effect of parent artery segmentation on intraneurysmal hemodynamics,” *American Journal of Neuroradiology*, vol. 27, pp. 1703–1709, 2006.

- [97] S. N. Wright, K. M. Brown, J. R. Cebral, P. Kochunov, J. C. Mazziota, S. L. Senft, A. W. Toga, and G. A. Ascoli, “Digital reconstruction and morphometric analysis of human brain vasculature from magnetic resonance angiography,” in *Society of Neuroscience Annual Meeting*, 2007.
- [98] S. Appanaboyina, F. Mut, R. Lohner, C. M. Putman, and J. R. Cebral, “Computational fluid dynamics of stented intracranial aneurysms using adaptive embedded unstructured grids,” 2007, *International Journal for Numerical Methods in Fluids*.
- [99] L. Antiga, B. Ene-Iordache, and A. Remuzzi, “Centerline computation and geometric analysis of branching tubular surfaces with application to blood vessel modeling,” in *Visualization and Computer Vision*, 2003.
- [100] S. Appanaboyina, F. Mut, R. Lohner, C. Putman, and J. R. Cebral, “Patient specific modeling of intracranial aneurysm stenting,” in *SPIE Medical Imaging Conference*, vol. 6511, 2007, pp. 65 112D–1 – 65 112D–11.
- [101] S. Appanaboyina, M. A. Castro, R. Lohner, and J. R. Cebral, “Simulation of endovascular interventions of cerebral aneurysms: Techniques and evaluation,” in *SPIE Medical Imaging Conference*, vol. 5746, 2005, pp. 769–779.
- [102] D. F. Kallmes, Y. H. Ding, D. Dai, R. Kadirvel, D. A. Lewis, and H. J. Cloft, “A new endoluminal, flow-diverting device for treatment of saccular aneurysms,” *Stroke*, vol. 38, pp. 2346–2352, 2007.
- [103] F. Calamante, P. J. Yim, and J. R. Cebral, “Estimation of bolus dispersion effects in perfusion mri using image-based computational fluid dynamics,” *NeuroImage*, vol. 19, pp. 342–352, 2003.

Curriculum Vitae

Fernando Mut was born on November 6th, 1976 in La Plata in the Province of Buenos Aires, Argentina. He finished his undergraduate studies in Physics at the *Instituto Balseiro*, Argentina in 2002. He also received a Master's degree in Physics at the *Instituto Balseiro* in 2003. The Master's thesis dealt with the simulation of multiphase flows with free surface using the level set method in stabilized finite elements.

Upon completion of the Master's program, he enrolled in the Computational Sciences and Informatics Ph.D. Program at George Mason University. During the first two years he took courses in computational fluid and solid dynamics. Additionally, in August 2007, he enrolled in the Mathematics Master's Program where he took courses in functional analysis, partial differential equations, and iterative methods for sparse linear systems. He obtained his Master's degree in Mathematics in June 2008.

During the last two years he has been working on improving the computational hemodynamics modeling of cerebral aneurysms. This work resulted in the present dissertation: "Extensions to the computational hemodynamics modeling of cerebral aneurysms", which was completed in August 2008.