

**A thesis submitted to The University of Manchester for  
the degree of PhD in Mechanical Engineering in the  
Faculty of Engineering and Physical Sciences**

# **Developing Tools for Simulating Biological Flows**

Joseph Dawes

2014

Mechanical Aerospace and Civil Engineering

# Abstract

The long term aim of this work is to investigate and develop efficient methods to analyse systems involving biological flows with fluid structure interaction, particularly cerebral aneurysms. Cerebral aneurysms are extremely complex biological structures and this work develops some of the groundwork required to eventually build the capability to efficiently simulate them computationally. This will enable the prediction of a patient specific risk model to aid in surgical decisions on a day to day basis or in large scale studies.

Fibrous immersed dynamic structures of cerebral aneurysms seem suited to immersed boundary method simulations. Because of this numerous Immersed Boundary methods are investigated to determine if they present a viable approach, and which of these is best suited.

A number of different computational tools have been built and investigated using FFTW, Eigen, OpenMP, and GNU OCTAVE to meet the requirements to allow numerous implicit and explicit approaches to be investigated. These tools are validated and tested for a number of cases. Preliminary work is also presented aimed at generating physically representative numerical models from MRI scans.

# **Declaration**

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or institute of learning.

# Copyright

1. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.
2. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.
3. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

---

4. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see <http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487>), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see <http://www.manchester.ac.uk/library/aboutus/regulations>) and in The University's policy on Presentation of Theses.

# Contents

<b>I. CFD simulation of Cerebral Aneurysms</b>	<b>8</b>
1. Introduction and overview	9
2. Literature review	12
3. The Navier Stokes equations	16
3.1. Rheology . . . . .	17
3.2. Flow regime . . . . .	19
4. Simulation setup	20
4.1. Geometry extraction . . . . .	20
4.2. Preliminary study design . . . . .	21
4.3. Mesh and boundary conditions . . . . .	22
4.4. Flow fields and grid independence . . . . .	24
5. Conclusions	27
5.1. Conclusions and approach . . . . .	27

<b>II. Immersed boundary methods</b>	<b>29</b>
<b>6. Problem definition</b>	<b>31</b>
<b>7. Literature review and method description</b>	<b>35</b>
7.1. Delta function definition . . . . .	35
7.1.1. Narrow support delta function . . . . .	38
7.1.2. Wide support delta function . . . . .	41
7.2. Continuous forcing approaches . . . . .	45
7.2.1. The immersed boundary method . . . . .	45
7.2.2. The Virtual Boundary method . . . . .	51
7.2.3. Discrete forcing approaches . . . . .	52
7.2.4. Ghost cells and Reconstruction techniques . . . . .	55
7.3. The immersed interface method . . . . .	57
7.3.1. Lattice Boltzman - Immersed Boundary method . . . . .	59
7.4. Conclusions . . . . .	61
<b>8. Solver design</b>	<b>62</b>
<b>9. Baseline numerical scheme.</b>	<b>63</b>
9.1. Discretisation in space . . . . .	70
<b>10. Linear Solvers</b>	<b>74</b>
10.1. Spectral approach . . . . .	75
10.2. Iterative approach . . . . .	78
10.2.1. Gauss and Jacobi solver . . . . .	78
10.2.2. Multigrid Solver . . . . .	79

10.3. Comparison . . . . .	81
10.4. Fractional Step methods . . . . .	82
10.4.1. Timestep limitations . . . . .	89
<b>11. Fluid solver validation</b>	<b>94</b>
<b>12. Processing complex boundaries</b>	<b>97</b>
<b>13. Validation and test cases</b>	<b>100</b>
13.1. Driven rotational flow in a cavity . . . . .	100
13.2. Taylor Couette flow . . . . .	101
13.3. Perturbed droplet in stationary flow . . . . .	103
13.4. Flow over a static circular cylinder . . . . .	107
<b>14. Limitations and alternative approaches</b>	<b>119</b>
14.1. Implicit solution . . . . .	120
14.2. Solution strategy with the flow solve operator . . . . .	122
<b>15. Further work</b>	<b>125</b>
<b>A. Aneurysm simulations using commercial codes</b>	<b>127</b>
A.1. 3D Chimney stent . . . . .	133
<b>Bibliography</b>	<b>137</b>



# List of Figures

1.1. Schematic showing the interconnected blood vessels making up the circle of Willis at the base of the brain. . . . .	10
4.1. Extracted surface showing the aneurysm location (a) and close up of aneurysm region (b) . . . . .	21
4.2. Medical scans used in cerebral aneurysm model construction . . . . .	22
4.3. Generated mesh and construction lines . . . . .	22
4.4. Meshes for the healthy (a), developing aneurysm (b) and developed aneurysm (c) cases . . . . .	23
4.5. Mean velocity physiological waveform . . . . .	24
4.6. Healthy geometry velocity vectors and pressure contours (note that the pressure divided by the constant density) . . . . .	25
4.7. Healthy geometry velocity vecotors and pressure contours (note that the pressure divided by the constant density) . . . . .	26
4.8. Healthy geometry velocity vectors and pressure contours (note that the pressure divided by the constant density) . . . . .	26

6.1. The fluid ( $\Omega_f$ ) and solid ( $\Omega_s$ ) domains and the interface ( $\Gamma$ ) between them in a general fluid structure interaction problem . . . . .	32
6.2. Meshes that conform (left) and do not conform (right) to interfaces . . . .	33
7.1. Various key concepts of the Immersed Boundary method . . . . .	45
7.2. Mirroring method for applying no slip boundary conditions . . . . .	54
9.1. Positioning of the velocity and pressure components on the MAC grid . .	71
10.1. Discrete transform types used in the Eigenfunction expansion solution method [1] . . . . .	91
10.2. Various key concepts of the multigrid strategy including the restriction and prolongation operations . . . . .	92
10.3. The V-cycle Multigrid algorithm . . . . .	93
11.1. Boundary conditions for fluid solver validation case . . . . .	94
11.2. Centreline x-velocity plot for the fluid solver validation case - comparison with Ghia <i>et al.</i> [2] . . . . .	95
11.3. Velocity vectors and streamlines for driven cavity flow with $Re=100$ . . .	96
11.4. Velocity vectors and streamlines for driven cavity flow with $Re=400$ . . .	96
12.1. Sample raster image for processing . . . . .	98
12.2. Point extraction and smoothing/resampling . . . . .	98
12.3. Sample velocity vectors with overlaid pressure and streamfunction con- tours for arbitrary geometry with boundary velocity = $(1, 1)$ in a cavity ( $Re=1$ ) . . . . .	99

13.1. Velocity vectors for rotational flow in a cavity driven by a circular immersed interface . . . . .	101
13.2. Contours of streamfunction for rotational flow in a cavity driven by a circular immersed interface . . . . .	102
13.3. Centreline and diagonally sampled x-velocity across the domain and intersecting the immersed geometry for rotational flow in a cavity generated using immersed boundaries . . . . .	103
13.4. Velocity vectors for Taylor-Couette flow in a cavity driven using immersed boundaries . . . . .	104
13.5. Contours of streamfunction for Taylor-Couette flow in a cavity driven using immersed boundaries . . . . .	105
13.6. Centreline and diagonally sampled x-velocity across the domain and intersecting the immersed geometry for Taylor-Couette flow in a cavity driven using immersed boundaries . . . . .	106
13.7. Initial interface configuration and computational domain geometry and boundary conditions for the unsteady perturbed droplet case . . . . .	107
13.8. Velocity vectors and droplet interfaces showing the dynamic response of the droplet at 0, 0.15, 0.3, 0.45, 0.6, 0.75 and 0.9 seconds. . . . .	111
13.9. Response of the droplet major and minor axes on the 128x128 grid using the colocated finite difference scheme, note the slight downward slope . .	112
13.10 Response of the droplet major and minor axes on the 128x128 grid using the staggered (MAC) finite difference scheme . . . . .	113

13.11 Droplet volume assuming the interface to be elliptical during the simulation. The top plot is for the staggered discretisation and shows a significant increase in volume conservation properties compared to the colocated grid (bottom) . . . . .	114
13.12 Reynolds number vs Strouhal number for the current results, Lai <i>et al.</i> [3] and the experimentally derived fitting of Williamson <i>et al.</i> [4] . . . . .	115
13.13 Contours of vorticity around a static circular cylinder calculated using the immersed boundary method for Reynolds numbers of 40 (top), 100 & 400 (bottom) . . . . .	116
13.14 Velocity magnitude around a static circular cylinder calculated using the immersed boundary method Reynolds numbers 40,100,150,200 and 400 increasing top to bottom. The immersed cylinder geometry is lightened for clarity . . . . .	117
13.15 Lift and drag coefficient for $Re = 150$ case . . . . .	118
A.1. Schematic showing the blood vessels connected to the aorta. . . . .	128
A.2. Geometry and boundary conditions for early axisymmetric AAA simulations . . . . .	129
A.3. Axisymmetric AAA geometry used in early studies . . . . .	129
A.4. Centreline velocity plot for 2 dimensional abdominal aortic aneurysm case	131
A.5. Wall shear stresses for abdominal aortic aneurysm case . . . . .	132
A.6. Pressure contours (kPa) for no stent and quarter deployed stent case . . .	133
A.7. 3D chimney stent and aorta geometry used in ANSYS simulations . . . .	134
A.8. Coarse mesh for the chimney stent investigation . . . . .	135

A.9. Velocity vectors and pressure contours in a chimney stent and untreated aortic aneurysm at highest flow rate . . . . .	135
A.10. Velocity vectors and pressure contours in a chimney stent and untreated aortic aneurysm at the average flow rate . . . . .	136
A.11. Velocity vectors and pressure contours in a chimney stent and untreated aortic aneurysm during flow reversal . . . . .	136

# List of Tables

3.1. Typical parameters for blood using the Carreau model . . . . .	18
9.1. Matrix representation of the Laplacian for various boundary conditions . .	72
10.1. Computational time requirement for various linear solution strategies for $2^n \times 2^n$ grid points . . . . .	81
13.1. Percentage area loss of the droplet during the first 4 seconds of the simu- lation . . . . .	107
13.2. Strouhal numbers calculated from the current results, Lai <i>et al.</i> [3] and the experimental results of Roshko <i>et al.</i> [5] . . . . .	110
A.1. Typical flow through the aorta and connecting arteries (from [6]) . . . . .	130

# Motivation

The motivation of this work is to explore, simulate and develop tools for investigating fluid and structural problems which are encountered in real life physiological systems. To handle complex geometries much development is required to assemble a comprehensive and robust approach, especially when strongly coupled interactions between fluids and structures are present. Immersed/Embedded geometry methods are investigated with the aim of assessing their capability for complex geometries and Fluid Structure Interaction (FSI) problems. Fluid Structure Interaction is of significant importance across a range of biological applications, such as in the modelling of blood flow, at a number of length scales, from the deformation of blood cells, thrombosis formation, the deposition or build up of atherosclerotic plaque and the flow/pressure induced deformation of vasculature. Cerebral and Aortic aneurysms are an early focus of this study. Aneurysms are physiologically important cases where advances in numerical simulations could prove potentially valuable, for example leading to improvements in treatment, pathogenesis or prevention.

**Part I.**

**CFD simulation of Cerebral  
Aneurysms**

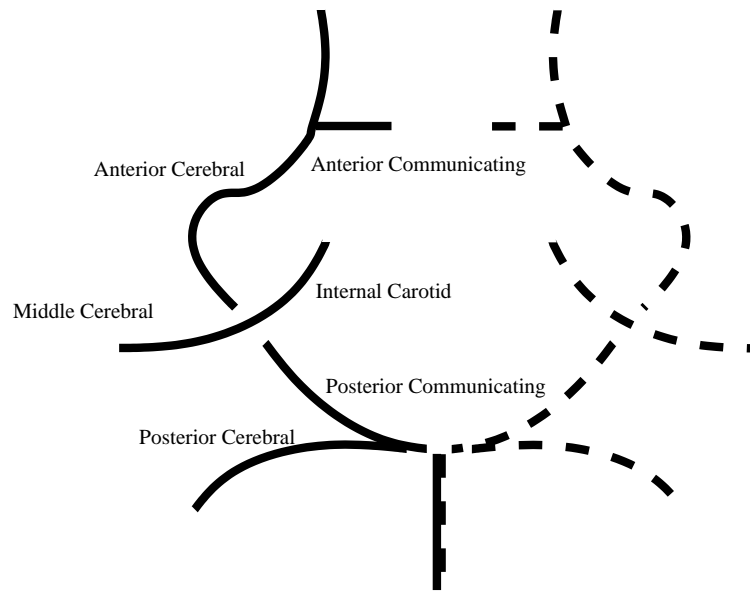


# 1. Introduction and overview

A case where there is much need for a better platform for numerical simulation is that of cerebral aneurysms. Cerebral aneurysms are local dilations in blood vessels towards the base of the brain, i.e. a section of blood filled vessel, supported and surrounded by intercranial fluid that has expanded due to pathogenic factors such as high blood pressure or lesions. They are typically saccular or fusiform in shape and occur near bifurcations within the Circle of Willis (Shojima *et al.* [7, 8]) which is a network of arteries which deliver blood to the brain. A schematic of the Circle of Willis is shown in Figure 1.1. The geometry of this network of arteries varies from patient to patient. Atypical topologies, such as those with blood vessel malformations, lead to a higher risk of aneurysm formation.

Cerebral aneurysms are relatively common, occurring unruptured in approximately 6% of the population [7]. Many of those diagnosed will not rupture, however those that do have a dismal prognosis from the resulting sub-arachnoid haemorrhage. Complications can also occur without rupture due to the physical size of the aneurysm or a dislodged thrombus.

Once a cerebral aneurysm has been identified in a patient two major treatment options ex-



**Figure 1.1.:** Schematic showing the interconnected blood vessels making up the circle of Willis at the base of the brain.

ist, surgical clipping and endovascular coiling. Surgical clipping consists of performing a craniotomy to expose the aneurysm, then closing the base of the aneurysm with a clip. Coiling uses a catheter to deploy platinum coils within the aneurysm which cause thrombosis. Both these methods block off the blood flow preventing further growth. These treatments come at relatively high risk. A series of comparative studies into the relative risks of aneurysm treatment have been presented by Molyneux et al. [9] which evaluate each treatment method. It was concluded that after one year the chance of survival free of disability was greater for endovascular coiling (76% as opposed to 69%). A later study of the same patients set five years later concluded that ruptured aneurysms treated with coiling had a slightly higher risk of recurrent bleeding compared to those treated with clipping, although in line with the earlier studies the overall the risk reduction was greater for coiling.

For unruptured aneurysms, it is often particularly challenging to balance the high risks associated with surgical intervention against the risks of leaving a particular aneurysm untreated. This is particularly difficult for aneurysms between 2 and 4 millimetres in diameter.

## 2. Literature review

There are a number of different approaches in literature geared towards pathogenic factors and quantifying risks.

Firstly, there are studies which ignore the details of the flow field, such as that of Hatiboglu et al. [10], who developed a risk criteria using fuzzy logic approach based on parameters such as patient age and blood volume. Cohort studies such as Molyneux et al. [9] represent a similar approach.

Details of the flow field can be found via CFD analysis. CFD simulations in simplified geometries are useful for determining general trends, correlating parameters such as wall shear stress magnitude or oscillation index to risk factors or areas typical of aneurysm formation. Examples of studies of this kind include the works of Alns et al. [11] and Paal et al. [12]. Alns et al. [11] simulated the flow within an empirically patient specific three dimensional Circle of Willis, investigating vessel radii and branch angles. High wall shear stresses were found in areas typical of aneurysm development. Pal et al. [12] simulated hemodynamic stress and flow fields in simplified (two and three dimensional) and real side wall intracranial aneurysm models aiming to draw parallels between the simplified and real geometries and noting geometries in which lower wall shear stresses are found.

Beyond simplified approaches such as these, efforts have been made to correlate detailed patient specific geometries and flow field information to cerebral aneurysm growth/formation or rupture risk. Patient specific geometries allow simulations to be used both in treatment selection, and to anchor correlations such as wall shear stress to growth and rupture (as described in the previous paragraph) to cases in which the outcomes are known, or the surgical decision has been made.

Medical imaging is currently unable to provide adequately resolved velocity fields [13] therefore CFD simulations are a vital tool to bridge this gap and allow the measurement of important features, such as impinging jets or wall shear stresses.

In CFD studies which neglect fluid structure interaction a number of different geometries and flow conditions can feasibly be completed and compared allowing for broader studies with detailed comparisons and correlations. Studies such as these include the work of Shojima et al. [7, 8], who note that, as simulated bloodstream impacting forces typically accounted for pressure rises of the order of 1%, there is no strong correlation with rupture or growth. In earlier work simulations of 20 patient specific MCA aneurysm simulations found a relatively weak negative correlation between aspect ratio and wall shear stress. Many of the lower aspect ratio aneurysms in the study were known to have later ruptured. Jou et al. [14], investigating hemodynamic forces in two giant cerebral aneurysms, noted that the geometry of the feeding vessels has a very significant effect on the hemodynamic forces which is a factor often left out of numerical studies.

The growth of cerebral aneurysms over large timescales has also been examined. The method and approach of these studies varies, but typically small timescale FSI effects are ignored. Paal et al. [12] investigated growth by calculating the blood flow using seven patient specific geometries which were extracted from MRI scans taken at 12 month in-

tervals. This work further strengthened the correlation between low wall shear stress and local aneurysm growth.

Chatziprodromou et al. [15, 16] used a fully computational approach by incorporating a model for wall remodelling and vascular tone change. An idealised geometry was used to simulate the development of a side wall aneurysm with a simple model for remodelling. A series of quasi-steady simulations were undertaken with an initial injury to the vessel wall, then remodelling was introduced into the simulation by setting the aneurysm wall to an unstressed state after each evolution step.

Figuerola et al. [17] present a detailed insight into how large timescale cardiovascular simulations can be undertaken. Growth was modelled by performing relatively short timescale simulations incorporating fluid structure interactions (using a coupled momentum method). An approximate hemostatic state was found and the vasculature remodelled over a large time scale. Growth was initiated by the removal of elastin in appropriate regions of the domain. Valencia et al.[18] [19] ran numerous patient specific simulations, finding a strong linear relationship between the aneurysm surface index, defined as the ratio between the aneurysm area and the artery area at the model inlet, and the spatially averaged WSS (calculated via CFD simulation) at peak systole.

Studies that incorporate the influence of FSI tend to be less broad in scope due to the computational requirements. In investigating the effects of aneurysm shape and wall compliance, Torii et al. [20] found a strong dependence on shape and influence of fluid structure interaction. The solution was very resource intensive, with one cardiac cycle taking approximately three days to solve using 16 cpus. These findings are backed up by earlier CFD FSI analysis by the same author ([21] (a linear elastic model was used for the vessel wall) on two patient specific vascular models which also found large differences in

WSS predictions with FSI. This is to be expected if the wall is in motion throughout the simulation. Valencia et al. [19] studied hemodynamics, deformation and effective stress within a healthy basilar artery and terminal basilar artery model incorporating FSI. The vessel wall thickness and constitutive model was found to have a significant impact on the results.

Further details of arterial aneurysm biomechanics can be found in Lasheras [22], a review concerning pathogenesis, growth and rupture of both aortic and cerebral aneurysms and a summary of active areas of research : pathogenesis, enlargement factors, quantifying rupture risk, and the development of patient specific treatments. The study emphasises two significant obstacles slowing the development and applicability of computational models, firstly, determining its mechanical properties and remodelling process of the arterial wall, which is a complex structure, and the lack of resolution in geometry and flow field reconstruction in current medical imaging techniques. Humphrey et al. [23] also provides a review of literature on the structure, mechanical properties and mechanics of intracranial aneurysms. Much of the work reviewed has since been superseded, however the paper serves as an excellent introduction to the subject.

Some improvements, such determining the aneurysm and surrounding vessel's topology and flow conditions are feasible to the limit of medical images, others such as growth over mixed time-scales, atherosclerotic plaque (Tateshima et al.) [24] and the complex patient specific vessel wall structure add a large amount of complexity and require data that is typically not available and/or varies on a case to case basis. Fluid Structure interaction and auto-regulation within the skull will have a strong impact and a broader applicability.

### 3. The Navier Stokes equations

In this work the solution of the Navier Stokes equations, consisting of the conservation of momentum, mass and energy are considered. The fluid is considered to be incompressible and as such the density will set to unity. The momentum equation is defined below in Equation 3.1

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{u}\mathbf{u} = \nabla \cdot \mathbf{T} \quad (3.1)$$

For a Newtonian fluid the stress tensor  $\mathbf{T}$  in Equation 3.1 can be expressed using Equation 3.2 shown below

$$\mathbf{T} = \left( p + \frac{2}{3}\mu \nabla \cdot \mathbf{u} \right) \mathbf{I} + 2\mu \mathbf{D} \quad (3.2)$$

$$\mathbf{D} = \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \quad (3.3)$$

Equation 3.3 gives the rate of strain tensor  $D$ . For an incompressible fluid  $\frac{2}{3}\mu \nabla \cdot \mathbf{u} = 0$  due



to the continuity equation, so the second term in the brackets in Equation 3.2 disappears.

These expressions result in the the following expression for the conservation of momentum and mass for an incompressible Newtonian fluid.

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{u}\mathbf{u} = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f} \quad (3.4)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (3.5)$$

## 3.1. Rheology

In complex fluids such as blood, the viscosity cannot always be considered constant. Blood consists of plasma and particles. Plasma is a Newtonian fluid mostly made up of water, dissolved proteins and glucose. Approximately 99% of the particulate volume of blood consists of red blood cells. The remaining particulate volume contains white blood cells (leukocytes and platelets).

At small length scales and shear rates, red blood cells can have a significant influence on the blood rheology, hence the hemocrit and the size, shape and deformability of the red blood cells, along with factors such as blood temperature are important in determining the viscosity of a blood sample ([25]). The presence of red blood cells makes blood shear thinning ([25, 26]), i.e. the apparent viscosity decreases as the shear rate increases. This is caused by the realignment of blood cells in high shear rate flow ([25]. )

Aggregation of red blood cells at low shear rate results in blood having a yield stress. In larger vessels, which have relatively high shear rates, viscosity can be assumed constant

### 3.1 Rheology

---

as the viscosity plateaus out. At the capillary level, or in channels not much wider than the diameter of a red blood cell, red blood cells show axial accumulation, i.e. bolus flow develops, and laminar/parabolic flow profiles are unrealistic ([27, 28].) The selection of a suitable and efficient rheological/constitutive model depends on the flow conditions.

Depending on the local shear rates, blood can behave in a close to Newtonian fashion, with a constant viscosity in the range of 3-4 mPas. For more complex geometries with high shear gradients present the viscosity will vary significantly over the geometry, especially near walls.

The non-Newtonian Carreau model, originating from Carreau et al., is commonly used for blood in numerical simulations. Typical parameters for the Carreau model, Equation 3.6 reproduced from [29] are displayed in Table 3.1

$$\eta = \eta_{\infty} + (\eta_0 - \eta_{\infty}) \left[ 1 + (\lambda \dot{\gamma})^2 \right]^{\frac{n-1}{2}} \quad (3.6)$$

$\eta_{\infty}$	0.056 Pas
$\eta_0$	0.00348 Pas
$\lambda$	3.31s
$n$	0.356

**Table 3.1.:** Typical parameters for blood using the Carreau model

### 3.2. Flow regime

Classically blood flow is considered laminar. The Reynolds numbers in blood vessels range from below 0.001 in small capillaries, to a peak value of approximately 5000 in the aorta during exercise.

Tanaka et al.[30] recorded total blood volume flow rates of approximately 800ml/min with 30% of the flow travelling through the MCA. Vessels around the MCA (see Figure 1.1) are in the range of 3mm in diameter, so the Reynolds numbers for cerebral aneurysm cases is typically around 100, although due to the range of flowrates and Circle of Willis topologies between patients it is feasible that Reynolds numbers could be higher. Flow will transition from laminar to turbulent in straight pipes between Reynolds numbers of 2300 and 4000 (Fox et al. [31]). Much of the circulatory system does not consist of straight tubes, particularly the areas in which aneurysms typically form. Stehbens et al.[32] found that turbulence was encountered at bends, bifurcations or unions at significantly lower Reynolds number than for straight channels. However, at typical Reynolds numbers, assuming laminar flow in cerebral aneurysms will be a relatively safe assumption. Additionally, Alnaes et al. [11] report that most blood vessel geometries in the body are formed in such a way as to minimize work, so the Reynolds numbers at which turbulence occurs predicted by Stehbens are likely underestimates for hemodynamic flows. In aortic aneurysms turbulence is a significant feature.

## **4. Simulation setup**

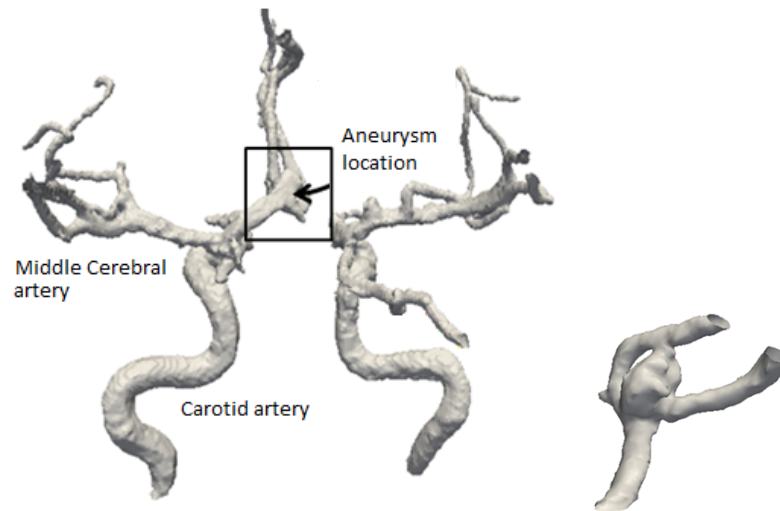
### **4.1. Geometry extraction**

Three dimensional simulations of a saccular cerebral aneurysms have been investigated as part of an initial collaborative study with the Neurosurgery Department at Salford Royal Hospital. The aneurysm chosen for this study is shown below. It is located before the bifurcation on the middle carotid artery.

The images shown above are surfaces extracted from rotational digital subtraction angiograph image provided by Salford Royal Hospital. Figure 4.1 shows slices of these images.

It is also interesting to note that in this Circle of Willis, the anterior communicating arteries are not developed. Malformations such as these significantly increase the risk of aneurysm formation.

Due to the relatively low resolution of the DICOM images used as a base, the extracted surfaces contain a number of artifacts, and the position of the vessel walls is only defined to a tolerance. It was decided to construct a smooth idealised geometry using ANSYS

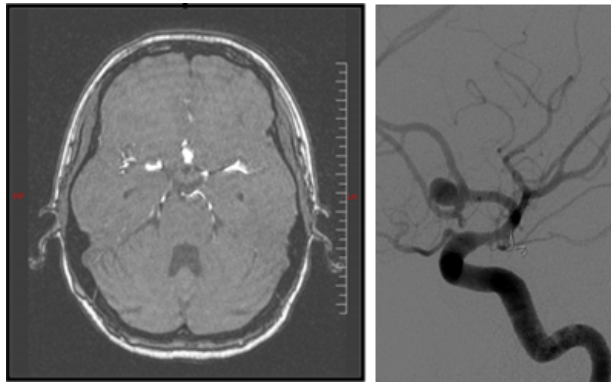


**Figure 4.1.:** Extracted surface showing the aneurysm location (a) and close up of aneurysm region (b)

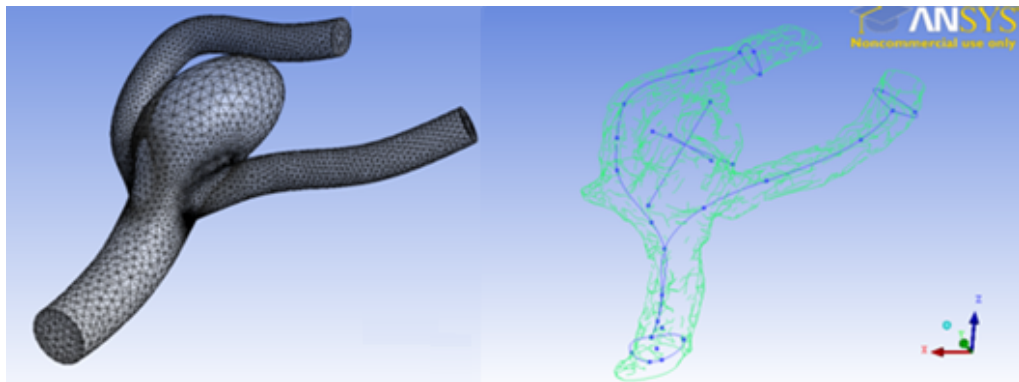
DesignModeller. Measurements of the vessel and sac diameters and lengths before and after the bifurcation and a number of centrelines were calculated from the STL surface in order to reconstruct the geometry in a manner better suited to CFD simulation. The reconstructed middle cerebral artery and connecting geometry could practically be more accurate than the extracted STL surface due to the artifacts inherent in the feature extraction.

## 4.2. Preliminary study design

This section compares the flow fields and wall shear stresses for a healthy vessel, a small aneurysm and a large fully developed aneurysm, a selection from a number of case studies. Firstly a grid density required for mesh independence was determined for the cerebral aneurysm geometry.



**Figure 4.2.:** Medical scans used in cerebral aneurysm model construction



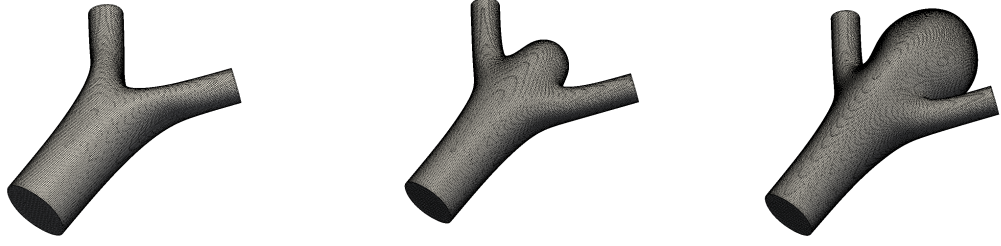
**Figure 4.3.:** Generated mesh and construction lines

## 4.3. Mesh and boundary conditions

Each geometry is displayed below in Fig.4.4

The MCA inlet is defined as circular with a diameter of 3mm, the bifurcating vessels downstream in each case have a diameter of 1.5mm, matching those of the patient specific study.

For steady state solutions fully developed pousielle flow is assumed in the vessel up-



**Figure 4.4.:** Meshes for the healthy (a), developing aneurysm (b) and developed aneurysm (c) cases

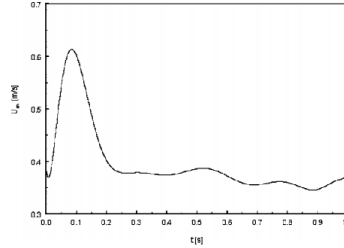
stream, a parabolic velocity profile of the form shown in 4.1 is defined.

$$v(r) = 2 * v_{ave} \left( 1 - (r/R)^2 \right) \quad (4.1)$$

For the unsteady case a Womersley (1955) profile is more reflective of the physiological conditions, however, it was found that by extending the domain upstream and fitting the same parabolic profile as in the steady case (but with a varying average velocity) a Womersley type profile was recovered without the need for a relatively complex implementation. The transient mean velocity physiological waveform used in this study is taken from Valencia et al. [19]. This waveform is reproduced as the figure below for convenience, Fig.4.5 As the waveform is defined in the Basilar artery a correction factor is used, which is calculated using the volume flow ratios measured by Tanaka et al..[30] Tanaka et al found that the volume flow ratio of the MCA is approximately 30%.

The steady state simulations use SIMPLE scheme for pressure velocity coupling and second order upwinding for the convective terms. Transient simulations use a PIMPLE scheme, which is a merged PISO-SIMPLE scheme allowing for large timesteps, although

the solution is advanced with the maximum Courant number limited to unity to ensure a reasonable solution.



**Figure 4.5.:** Mean velocity physiological waveform

## 4.4. Flow fields and grid independence

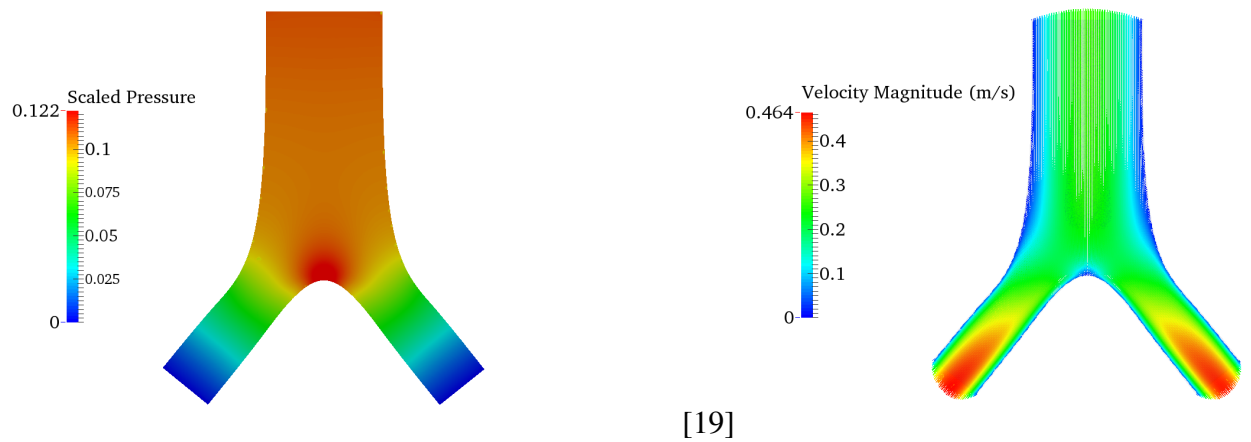
The aneurysm geometry shown in Fig.4.4 was meshed using successively refined grids. Both the error in peak wall shear stresses and pressure losses over the domain changed by less than 1% for the two finer grids. The medium grid, with 200,000 cells produced results to within 5% of the others.

The fine grid is a good balance between computational time and accuracy, further refinement beyond this point makes little practical difference.

The pressure contours and velocity vectors for the healthy case, the highest pressure and shear stress are at the joint of the bifurcation where flow is being redirected into the bifurcating channels.

Pressure contours and velocity vectors for a smaller aneurysm show a small area of recirculation and large area of increased pressure where the flow is stagnating against the vessel wall. The peak wall shear stress in this case is 8.19 Pa, remaining low in the aneurysm





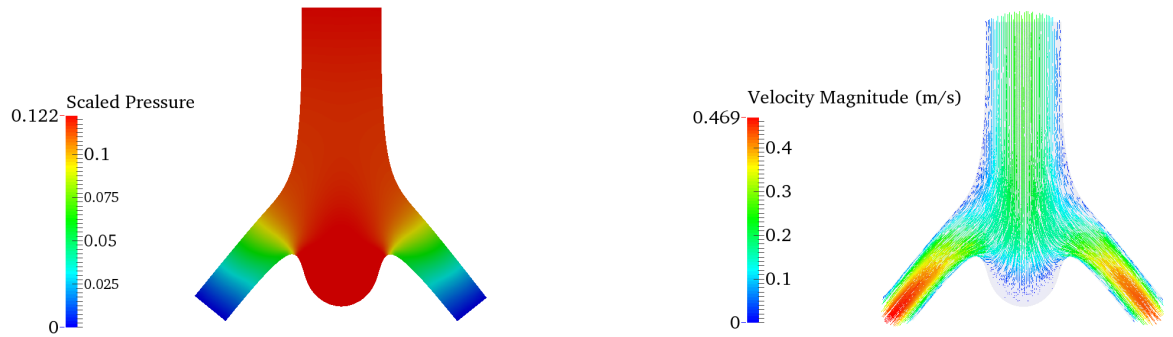
**Figure 4.6.:** Healthy geometry velocity vectors and pressure contours (note that the pressure divided by the constant density)

body, the mean wall shear stress is 0.29 Pa in the aneurysm body. It is significantly lower than the healthy vessel.

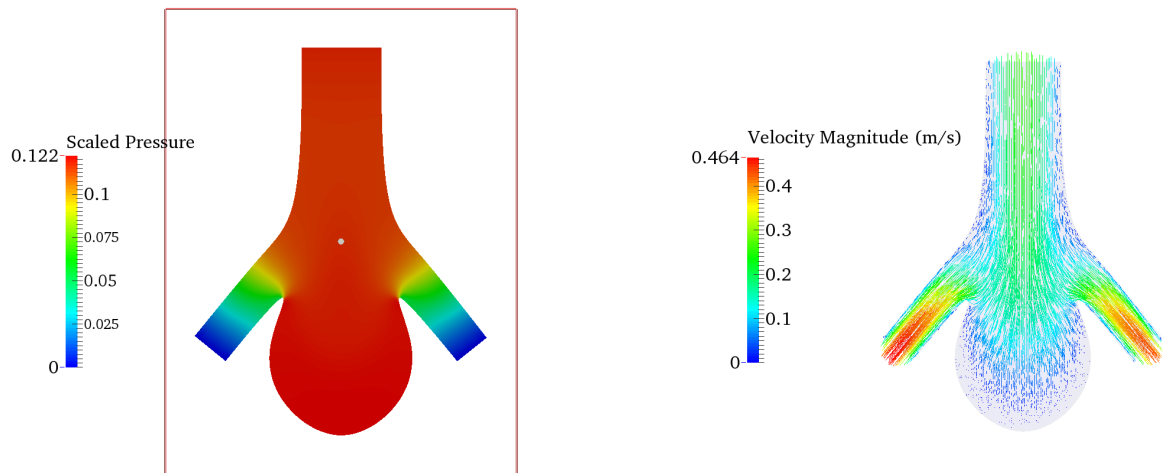
A giant aneurysm exhibits a larger zone of recirculating flow and stagnant flow. The peak wall shear stresses occur where the bifurcating vessels meet the aneurysm body.

#### 4.4 Flow fields and grid independence

---



**Figure 4.7.:** Healthy geometry velocity vecotors and pressure contours (note that the pressure divided by the constant density)



**Figure 4.8.:** Healthy geometry velocity vectors and pressure contours (note that the pressure divided by the constant density)

## **5. Conclusions**

The primary objective of this work is to find a suitable approach to simulate and predict risk factors for cerebral aneurysms and outlined in the previous section. This section contains an overview and justification of the assumptions that are made in the selection of a candidate solution strategy.

### **5.1. Conclusions and approach**

For the purposes of this study, flow will be considered laminar. Further investigation is required to determine if a turbulence model will be beneficial in aortic aneurysm simulations. The inclusion of a basic turbulence model is a straightforward task and can be investigated after the more complex issue of fluid structure interaction.

Initially Newtonian fluids will be considered. Forces transmitted to the vascular material will be much more strongly dependent on pressure forces and geometry changes, although shear thinning will impact wall shear stresses and may well alter the flow fields resulting in flow impinging on different sections of vascular structure.

## 5.1 Conclusions and approach

---

Current approaches using commercial codes typically require the generation of body fitted meshes, requiring a large amount of user input, and the definition of a specific numerical and geometrical parameters to which only estimates are available, as the underlying geometry and material parameters can only be defined to a tolerance. For this reason it has been decided to investigate immersed boundary methods to determine their applicability to this task.

There are many approaches to addressing fluid structure interaction. Overall they can be categorised into two different groups, monolithic approaches, where the equations for the fluid flow and structural displacements are solved simultaneously, and partitioned approaches where the flow and structure are solved separately using different solvers.

The partitioned approach allows different discretisation and numerical algorithms to be used for the fluid and structural domains. This is beneficial as it allows the a broad choice of numerical tools to be selected, so that the best suited may be chosen for each domain.

## **Part II.**

# **Immersed boundary methods**

---

Immersed boundary methods are becoming a popular tool for fluid simulations in geometries where mesh generation would be prohibitive, either due to complex geometries or boundary movement. The immersed boundary method (as opposed to other methods for the solution of flows containing immersed boundaries, of which it was also one of the first of its kind) was originally developed by Peskin [33] in 1972 in order to simulate blood flow in heart valves and numerous other biological applications where there is need to handle immersed elastic membranes. Problems of this kind are commonly found in biophysical flows. Immersed boundary methods, as stated in a comprehensive review by Mittal et al., [34] can fall into two categories based on at which point the boundary forces are introduced to the Navier Stokes equations. Continuous forcing approaches add terms to the Navier-Stokes equations before discretisation. This group includes the method of Peskin, known as the 'Immersed Boundary Method', and the virtual boundary method. Discrete forcing approaches originated with the work of Mohd-Yusof. The general idea of discrete forcing approaches is to define a forcing term such that the velocity on the boundary is set to a prescribed value. There are a number of methods based on this approach, including Ghost Cell methods. Other discrete forcing approaches include Cut-Cell methods and the immersed interface method. A detailed description of each of these methods and an overview of the current state of the art is presented in chapter 7.

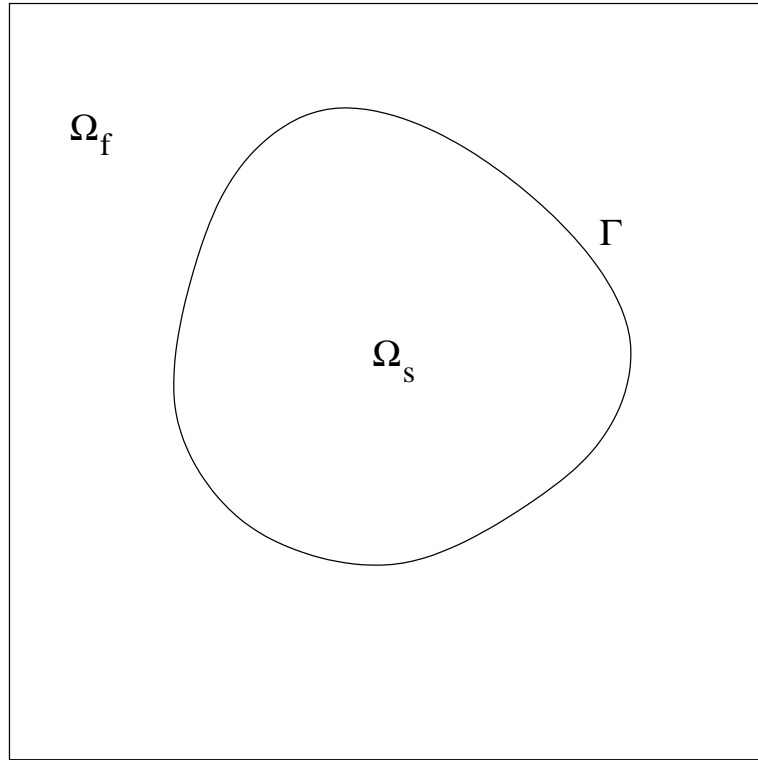
## 6. Problem definition

The immersed boundary methods to be investigated act on a domain containing fluid, which we will consider to be incompressible. Immersed within this domain, which will be referred to as  $\Omega_f$ , are one or more domains, interfaces  $\Gamma_n$ , or interfaces between domains which require special treatment. The Navier Stokes equations (chapter 3) are typically solved over the entire domain with local alterations to account for boundaries .

The special treatment consists of adding boundary conditions, constitutive models or otherwise altering the governing equations.

Depending on the problem under consideration, the domains to which special treatment will be applied will have equal or fewer dimensions than the fluid domain. In the case of equal dimensions, sub or separate domains  $\Omega_{sn}$  with interfaces between the domains, which will be referred to as  $\Gamma_n$  , may be considered. A lower dimensional object leaves only interfaces  $\Gamma_n$  between parts of the fluid domain for treatment.

The exact nature of the alteration to the governing equations depends on the problem under consideration, which may consist of modelling immersed elastic structures, assigning values on an interface (such as Dirichlet, Neumann or Robin boundary conditions in



**Figure 6.1.:** The fluid ( $\Omega_f$ ) and solid ( $\Omega_s$ ) domains and the interface ( $\Gamma$ ) between them in a general fluid structure interaction problem

Equation 6.1) or setting values over the entire domain.

$$A\nabla\phi + B\phi = A\nabla_{\Gamma}\phi_{\Gamma} + B\phi_{\Gamma} = C \text{ on } \Gamma \quad (6.1)$$

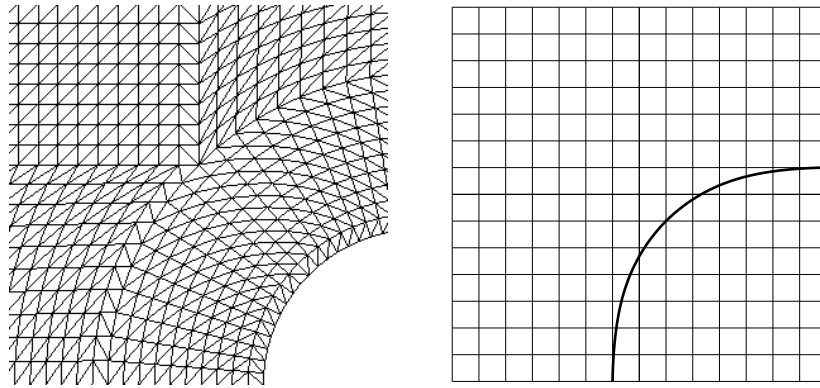
It is often the case that a value is prescribed to an interface immersed within the fluid domain in which case Equation 6.1 reduces to the form shown below.

$$\mathbf{u} = \mathbf{U} \text{ on } \Gamma$$



Where  $\mathbf{U}$  and  $\mathbf{u}$  are the velocities defined on the boundary and in the fluid domain respectively.

Traditional approaches require the mesh onto which the governing equations are discretised to conform to the edges of the boundary so that the interface can be treated as a physical boundary condition. Immersed boundary methods remove this restriction by imposing constraints or otherwise altering or adding to the model equations to represent the influence of the boundary. This allows the numerical discretisation to be considered independent of the geometry if required. Examples of conformal and non-conformal grids are displayed in Figure 6.2.



**Figure 6.2.:** Meshes that conform (left) and do not conform (right) to interfaces

If the surface is moving, a conformal grid must be remeshed in the vicinity of the surface and interpolation is required to transform between the old and new meshes. In the case of non-conformal meshes there still may be remeshing to allow for local grid refinement, although this will be less computationally expensive.

A frame of reference must also be specified for each domain. Conventionally fluids are considered in the Eulerian frame of reference (typically using finite volume or finite difference methods) and structures are modelled in the Lagrangian frame (often using finite

element methods).

## Non-Conformal methods

Adopting an Eulerian frame of reference for  $\Omega$ , where the coordinate frame is fixed in space, and a Lagrangian frame of reference for  $\Gamma$ , where coordinate system is convected with the material under consideration, allows the treatment of geometries or interfaces with complex shapes to be introduced and analysed in a straightforward manner in the Lagrangian frame.

Without physical boundary conditions in the Eulerian frame the solution of the Navier Stokes equations may be greatly simplified by, for example, utilising equally spaced Cartesian grids.

In almost every practical application the Lagrangian points  $\Gamma_h$  and the fixed coordinates of  $\Omega_h$  will not be coincident, so simplifications gained by considering two frames of reference come at the expense of additional interpolations due to the transformation of data between them.

The following section categorises and details approaches to solving and advancing problems of these kinds.

## **7. Literature review and method description**

### **7.1. Delta function definition**

Delta functions are a topic intertwined with immersed boundary methods. A description and derivation of commonly used models follows in this section.

A defining element of the immersed boundary method is the use of discrete delta functions. Discrete delta functions have the effect of smoothing the singular source terms from Equation 7.30 over a number of grid points, thus overcoming the fact that the grid points rarely coincide with the boundary points.

Equation 7.30 and Equation 7.31 contain terms linking the Eulerian and Lagrangian descriptions which must be conserved between the frames of reference. This leads to Equation 7.1 ([35]).

$$\int \delta(\Gamma, g, x) f(\mathbf{x}) dx = \int g(s) f(\mathbf{X}(s)) ds \quad (7.1)$$

Where  $\delta$  is a continuous delta function and  $g$  a function spanning both frames of reference.

This integral needs to be regularised over the domain by replacing it with a discrete sum.

For convenience delta functions are typically assembled from one dimensional functions.

These functions are applied to problems of multiple dimensions by either computing their product, (Equation 7.2) or using a distance function (Equation 7.3).

$$\delta_h(\Gamma, g, \mathbf{x}) = \int_{\Gamma} \prod_{k=1}^d \delta(\mathbf{x}^k - \mathbf{X}^k(S)) g(S) dS \quad (7.2)$$

$$\delta_h(\Gamma, g, \mathbf{x}) = g(\mathbf{x}) \delta_{\epsilon}(d(\Gamma, \mathbf{x})) \quad (7.3)$$

Here  $\delta_h$  is the discretised/regularised delta function.

Concentrating on delta functions of the type Equation 7.2, a delta function can be constructed.

$$\delta_h(\mathbf{x}) = \frac{1}{\Delta h^N} \prod_{i=0}^N \phi\left(\frac{\mathbf{x}_i}{r}\right)$$

The function  $\phi$  is constructed using a number of postulates. The first postulate is compact support, which is introduced primarily to reduce the computational cost of the solution. A

## 7.1 Delta function definition

---

large support will result in the convolution operations coupling a large number of Eulerian grid values to each Lagrangian marker.

$$\phi(x/h) = 0 \quad \text{for } (|r| \geq r_s) \quad (7.4)$$

Where  $h$  is the grid spacing.

Physically, mass, force and torque need to be consistent between coordinate systems. This results in a moment condition which can be defined up to a given order. This expression (for the case of  $r = 0$ ) also ensures that the discretised values sum to unity thus conserving values during spreading and interpolation operations.

$$h \sum_{j=-\infty}^{\infty} \delta_{\epsilon}(x_j - x) (x_j - x)^n = \begin{cases} 1, & r = 0 \\ 0, & 1 \leq n < q \end{cases} \quad (7.5)$$

Finally the delta function needs to be translation invariant, it is only possible to introduce this postulate in a way such that it is independant of shifts diagonally i.e. where  $r_1 = r_2$  leading to the following postulate

$$\sum_{j=-\infty}^{\infty} \delta(x_j - x) \delta(x_j - x) = C \quad (7.6)$$

### 7.1.1. Narrow support delta function

On staggered finite difference or finite volume grids a relatively narrow stencil can be used. With the support of the delta function assigned to 3 grid points and the order of the moment condition  $q$  set to 1, Equation 7.4 and Equation 7.5 can be expressed at consecutive gridpoints within the function's support. The following algebraic approximation for  $\phi$  is determined.

$$\phi = \begin{cases} \frac{1}{3} \left( 1 + \sqrt{-3r^2 + 1} \right) & 0 \leq |r| < \frac{1}{2} \\ \frac{1}{6} \left( 5 - 3|r| - \sqrt{-3(1 - |r|)^2 + 1} \right) & \frac{1}{2} \leq |r| < \frac{3}{2} \\ 0 & \frac{3}{2} \leq |r| \end{cases} \quad (7.7)$$

### Derivation

Restricting  $r$  to between 0 and 1 allows us to construct the following equations using the postulates defined above (Equation 7.4, Equation 7.5 and Equation 7.6).

$$\phi(r-1) + \phi(r) + \phi(r+1) = 1 \quad (7.8)$$

$$(r-1)\phi(r-2) + (r)\phi(r-1) + (r+1)\phi(r+1) = 0 \quad (7.9)$$

## 7.1 Delta function definition

---

$$(\phi(r-1))^2 + (\phi(r))^2 + (\phi(r+1))^2 = C \quad (7.10)$$

Rearranging Equation 7.8 and substituting into Equation 7.9 -

$$\phi(r-1) = \frac{-\phi(r) + r + 1}{2} \quad (7.11)$$

$$\phi(r+1) = \frac{-\phi(r) - r + 1}{2} \quad (7.12)$$

Substituting Equation 7.11 and Equation 7.12 into Equation 7.10 -

$$\left(\frac{-\phi(r) + r + 1}{2}\right)^2 + (\phi(r))^2 + \left(\frac{-\phi(r) - r + 1}{2}\right)^2 = C \quad (7.13)$$

To find C, we rewrite Equation 7.13 in terms of  $\phi(r-1)$

$$\phi(r-1)^2 + (r+1 - 2\phi(r-1))^2 + (\phi(r-1) - r)^2 = C \quad (7.14)$$

When  $r = -\frac{1}{2}$ ,  $\phi(-\frac{1}{2} - 1) = 0$  due to the support of the delta function being defined as 3.

$$\left(\frac{1}{2}\right)^2 + \left(-\frac{1}{2}\right)^2 = \frac{1}{2} = C$$

Now, expanding and collecting like terms in Equation 7.14

$$\left(\frac{3}{2}\right)\phi(r)^2 + (-1)\phi(r) + \left(\frac{r^2}{2}\right) \quad (7.15)$$

The root of this equation is

$$\phi(r) = \frac{1}{3} \left(1 + \sqrt{1 - 3r^2}\right)$$

using this expression in Equation 7.12

$$\phi(r+1) = \frac{-\frac{1}{3} \left(1 + \sqrt{1 - 3r^2}\right) - r + 1}{2} \quad (7.16)$$

Simplifying

$$\phi(r+1) = -\frac{1}{6} \left(3r - 2 + \sqrt{1 - 3r^2}\right) \quad (7.17)$$



Transforming the variables

$$\phi(r) = -\frac{1}{6} \left( 3(r-1) - 2 + \sqrt{1 - 3(r-1)^2} \right) \quad (7.18)$$

$$\phi(r) = \frac{1}{6} \left( -3r + 5 - \sqrt{1 - 3(r-1)^2} \right) \quad (7.19)$$

### 7.1.2. Wide support delta function

If central differences are used to assemble the Poisson equation on a non-staggered grid, the situation can arise where the pressure field decouples as the poisson operator does not define a relationship between adjacent points. This problem only affects the pressure equation due to the lack of a diagonal component. If centred differencing is used to determine the velocity correction based on this decoupled pressure, the gradients use subsets of points from the decoupled field. When a reference pressure is set or assumed, these subsets are, within themselves, determinate. This means that the gradients still correctly remove the divergence from the velocity field and this potential instability does not spread to the momentum equation. When forcing is introduced it is important to expand the delta function's support as the pressure is effectively defined over a grid of double spacing. Also it is important to ensure that the forcing is spread over each of these decoupled grids

evenly which leads to the additional postulate

$$\sum_{j \text{ even}} \delta(x_j - x) = \sum_{j \text{ odd}} \delta(x_j - x) = 0.5 \quad (7.20)$$

With this postulate introduced and the support broadened to 4 gridpoints the following function for  $\phi$  is found.

$$\phi = \begin{cases} \frac{1}{8} \left( 3 - 2|r| + \sqrt{1 + 4|r| - 4r^2} \right) & 0 \leq |r| < 1 \\ \frac{1}{8} \left( 5 + 2|r| - \sqrt{-7 + 12|r| - 4r^2} \right) & 1 \leq |r| < 2 \\ 0 & 2 \leq |r| \end{cases} \quad (7.21)$$

## Derivation

Restricting  $r$  to between 0 and 1 allows us to write out the postulates as follows

$$\phi(r-2) + \phi(r) = \phi(r-1) + \phi(r+1) = \frac{1}{2}$$

$$2\phi(r-2) + \phi(r-1) - \phi(r+1) = r$$

$$(\phi(r-2))^2 + (\phi(r-1))^2 + (\phi(r))^2 + (\phi(r+1))^2 = C \quad (7.22)$$

Expressing the first three equations in terms of  $\phi(r)$

$$\phi(r-2) = \frac{1}{2} - \phi(r) \quad (7.23)$$

$$\phi(r-1) = \frac{r}{2} - \frac{1}{4} + \phi(r)$$

$$\phi(r+1) = -\frac{r}{2} + \frac{3}{4} - \phi(r)$$

for  $r = 0$ ,  $\phi(r-2) = 0$ ,  $\phi(0) = \frac{1}{2}$  and  $\phi(\pm 1) = \frac{1}{4}$ ,

substituting these values into Equation 7.22 determines  $C = \frac{3}{8}$ , from which we can derive

$$4(\phi(r))^2 - (3-2r)\phi(r) + \frac{1}{2}(1-r)^2$$

## 7.1 Delta function definition

---

Finally solving this quadratic equation gives

$$\phi(r) = \frac{3 - 2r + \sqrt{1 + 4r - 4r^2}}{8} \quad (7.24)$$

To find the value of  $\phi(r)$  for  $-2 \leq r \leq -1$  we first substitute equation Equation 7.23 into equation Equation 7.24.

$$\phi(r-2) = \frac{1}{2} - \frac{3 - 2r + \sqrt{1 + 4r - 4r^2}}{8} \quad (7.25)$$

The final expression is found by a change of variable

$$\phi(r) = \frac{1 + 2(r+2) - \sqrt{1 + 4(r+2) - 4(r+2)^2}}{8} \quad (7.26)$$

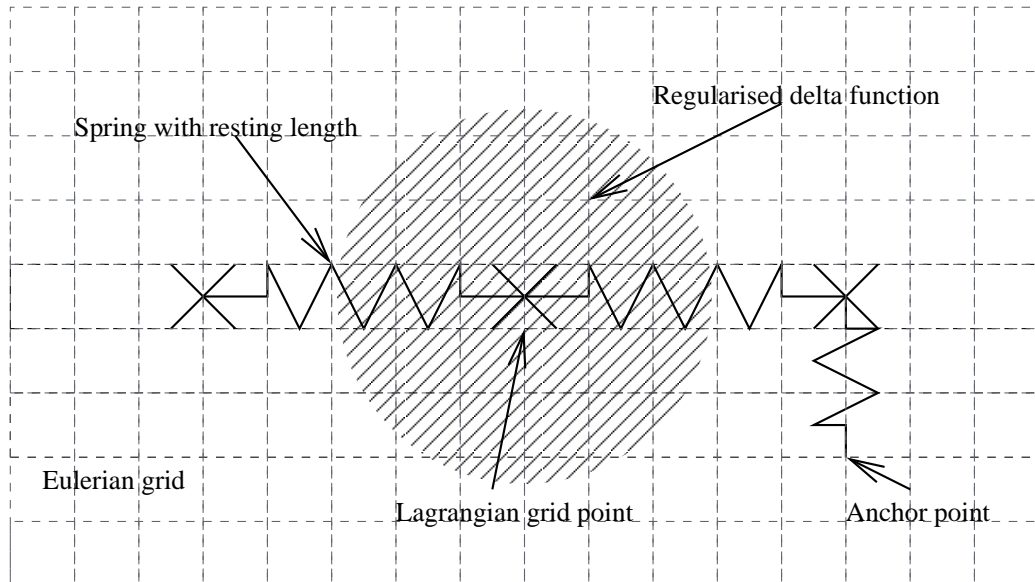
Simplifying this expression leads to

$$\phi(r) = \frac{5 + 2r - \sqrt{7 - 12r - 4r^2}}{8} \quad (7.27)$$

All that remains to complete this description is to discretise the solution in space and time which is discussed in chapter 3.

## 7.2. Continuous forcing approaches

### 7.2.1. The immersed boundary method



**Figure 7.1.:** Various key concepts of the Immersed Boundary method

The immersed boundary method is a widely used tool in the modelling of biological fluid dynamics where elastic fibres or membranes immersed within fluids are commonly encountered. Immersed boundary (IB) methods typically model a system including one or more interfaces (liquid-solid, liquid-liquid, or liquid-gas) using a grid that does not conform to the interface geometry.

The main benefits of such numerical implementations are the removal of the need to generate and, in the case of a moving interface, adapt body-fitted grids which, depending on the problem, can be a time consuming and numerically complex task.

An overview of many more developments in this approach and the numerous alternative

approaches for treating immersed boundaries is presented in Mittal *et al.* [34].

The general method described in this section roughly follows the descriptions given in most immersed boundary method papers (such as [36, 37, 38] etc).

The entire domain, both the boundary position and configuration and the fluid that it is immersed in, is governed by the incompressible unsteady Navier-Stokes equations with an additional momentum source term solved on a stationary Cartesian grid.

In the classical method, the immersed boundary is made up of sets of massless elastic fibres that are convected using the fluid velocity. These elastic fibres are discretised by Lagrangian marker points along their length, such that in the two dimensional case an immersed interface is represented by a curve parameterised along its length  $S$  of the form  $\mathbf{X}(s, t)$ . The extension to higher dimensions, as would be required for the treatment of membranes or interfaces with thickness, is straightforward.

The boundary configuration may generate a force (Equation 7.28) at the Lagrangian markers which is calculated using the interface's constitutive equation. Often Hooke's law is used, although any linear or non-linear relationship defining the material's constitutive law can be applied.

$$\mathbf{F}(s, t) = S(\mathbf{X}(s, t)) \quad (7.28)$$

Where  $\mathbf{F}$  is the force defined on the interface, and  $\mathbf{X}$  and  $s$  are the interface's coordinates and local coordinate system respectively.

The boundary force does not have to be defined by the configuration alone, it is often practical to introduce damping terms.

Alternatively the Lagrangian marker points making up the immersed interface (or a subset of these points) can be anchored to or driven to a fixed or moving position. This is often the case in simulations where it is required to apply forcing at the boundary to mimic a solid fixed interface - the interface is treated as an extremely stiff deformable structure. In the simplest case this results in the anchoring of each Lagrangian point defining the interface to its initial position using Hooke's law.

$$\mathbf{F} = k(\mathbf{X} - \mathbf{X}_0) \quad (7.29)$$

Where  $\mathbf{X}_0$  is the interface's initial configuration and  $k$  is the stiffness parameter.

This force is then applied back to the fluid.

Communication between the Lagrangian and Eulerian frames are handled by convolution expressions making use of a discrete delta function  $\delta_h$ , which is effectively a distance weighted kernel derived to meet a number of unique requirements which are outlined in the following section. In the continuous sense the spreading of the force generated by the interface is expressed by Equation 7.30.

$$\mathbf{f}(\mathbf{x}, s) = \int_{\Gamma} \mathbf{F}(s, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) ds \quad (7.30)$$

Where  $\mathbf{f}$  is the force density applied to the fluid.

The fluid is advanced with this force which has been spread through discretisation of the above expression. Often the fluid equations are discretised and solved on a fixed Cartesian grid, typically equally spaced although local grid refinement can be introduced.

It remains to advance the Lagrangian markers which make up the interface with the calculated velocity field. The convection of the Lagrangian field with the Eulerian velocity field is achieved in the continuous sense with the convolution expression Equation 7.31 below.

$$\mathbf{U}(\mathbf{X}, s) = \int_{\Omega} \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) d\mathbf{x} \quad (7.31)$$

Where  $\mathbf{U}$  and  $\mathbf{u}$  are the velocities of the interface and of the fluid in the Eulerian frame respectively.

Figure 7.1 graphically illustrates many of the concepts introduced above.

This approach has the capability of representing more complex immersed structures because two or three dimensional objects can be assembled through the superposition of numerous immersed fibres.

Since the inception there has been much work extending the original immersed boundary method ([38]). Notably there have been improvements in the temporal discretisation made allowing for formal second order method (i.e. a significant reduction in numerical diffusion allowing for second order convergence of the  $L^2$  norms) in Lai and Peskin [?].



The method has also been extended to incorporate local grid refinement in Roma *et al.* [39], Löhner *et al.* [40], boundary mass inertial effects [41], and fully implicit boundary advancement schemes (such as the Jacobian-free Newton–Krylov method) previously used with immersed interface techniques (see [42, 43]).

Bhalla *et al.* [44, 45, 46] present a number of test cases solved using the immersed boundary method with cartesian grid adaptive mesh refinement (AMR) allowing thin boundary layers at fluid–solid interfaces.

Chern *et al.* [47] used a direct-forcing immersed boundary method in a finite volume framework to investigate the evolution of oscillatory flow in arrays of cylinder structures to be efficiently captured. Test cases include Stokes’ first problem, a moving plate in an infinite flow, flow past a rigid cylinder, an oscillating rigid cylinder in an infinite flow, a falling sphere, an impulsively started cylinder and cylinders and spheres in shear driven flow. More complex two dimensional simulations of swimming fish are also presented.

Hu *et al.* [48] developed a simple immersed boundary method to simulate the dynamics of three-dimensional axisymmetric inextensible vesicles, a technique ideally suited to red blood cells. The vesicle’s near incompressibility is controlled using a spring like tension.

Wang *et al.* [49] present a parallel computing strategy for running immersed boundary method simulations with excellent parallel performance using a discrete stream-function formulation.

Borazjani *et al.* [50] use the Curvilinear immersed boundary method, incorporating overset-curvilinear grids to handle multi-connected geometries undergoing arbitrarily large deformations. Curvilinear grids allow for efficient refinement near boundaries. The work

focuses on ensuring globally conservative interpolation at grid interfaces suitable for incompressible flow fractional step method. The method is verified and validated against experimental data, and its capabilities are demonstrated by simulating the flow past multiple aquatic swimmers and the systolic flow in an anatomic left ventricle with a mechanical heart valve implanted in the aortic position.

The Immersed Structural Potential Method was recently introduced by Gil *et al.* [51], which shares many features with the immersed boundary method, modelling immersed solids by calculating the forces as the gradient of a deviatoric strain energy functional. An interesting feature of this work are the spline based kernels reducing/removing the spurious oscillations often found in immersed boundary methods. Test cases included contact of a flapping leaflet in pulsatile flow, the deformation of an elastic wall by flow and a deformable cylinder in a lid-driven cavity.

The immersed boundary method is well suited to the simulation of elastic structures especially in cases with large deformations, and it is straightforward to implement. The drawbacks of this method are that it requires a very small timestep to maintain stability, and the extension to more complex objects is not straightforward.

### 7.2.2. The Virtual Boundary method

Goldstein *et al.* [52] applied forcing to simulate solid boundaries using a feedback loop approach as shown by

$$\mathbf{f}(\mathbf{x}, t) = \alpha \int_0^t \mathbf{u}(\mathbf{x}, t') dt' + \beta \mathbf{u}(\mathbf{x}, t) \text{ for } \mathbf{x} = \mathbf{X} \quad (7.32)$$

Equation 7.32. Goldstein originally only considered boundaries where the boundary points and underlying mesh nodes were coincident, but it is a straightforward task to add smoothing, or interpolation and spreading operations to overcome this difficulty.

The immersed boundary method described above with the forcing calculated using Equation 7.29 is mathematically similar to the virtual boundary method with  $\beta = 0$  and Penalisation methods, where the entire domain is modelled as a porous medium using the Navier-Stokes-Brinkmann equations which are similar to the case where  $\alpha = 0$ .

Brown *et al.* [53] present several methods, including volume penalization methods in which boundary conditions are applied by using Brinkman penalization, special case of Equation 7.32 - representing solid obstacles as porous media. The paper extends this approach to incorporate homogeneous and inhomogeneous Dirichlet, Neumann and mixed/Robin boundary conditions and finds reasonable agreement for validation cases - flow around an adiabatic and a heated cylinder.

Introini *et al.* use a second order penalized direct forcing method to simulate fluid-structure interaction problems. Unlike Brown *et al.* [53], the boundaries were limited to Dirichlet

conditions. The second order linear interpolation scheme introduced is investigated and numerical rate of convergence of this method is quasi-quadratic. Taylor Couette solution, Poiseuille flow in an inclined channel and laminar flows around a cylinder and induced by a stirrer are presented.

Abgrall *et al.* [54] used a Penalization technique on unstructured meshes using mesh adaptation to refine near the immersed boundaries. The solution strategy is implicit, but limited to steady problems. The author notes a common drawback of this and other Penalisation methods - the introduction of a nonphysical penalisation parameter that has a significant impact on stability. Four test cases are considered, a simple Blasius test case, supersonic flow around a triangular obstacle and laminar flow around a NACA 0012 airfoil. The authors intend to extend the capabilities of the solver to incorporate moving bodies.

The virtual boundary method, although allowing more control for static boundaries than the immersed interface, has few applications where discrete forcing approaches would not prove more robust and effective.

### 7.2.3. Discrete forcing approaches

Discrete forcing approaches make use of the discretised terms in the momentum equation.

A forcing term can also be extracted from the momentum equation following the approach of Mohd-Yosuf [55] whereby the semi-discretised momentum equations are rearranged to find a force term to set the velocity at a given grid point using Equation 7.33. This is achieved by first advancing the Navier Stokes equations without incorporating the effects

of the boundary, then using this prediction to calculate a force with which to assign the boundary velocity.

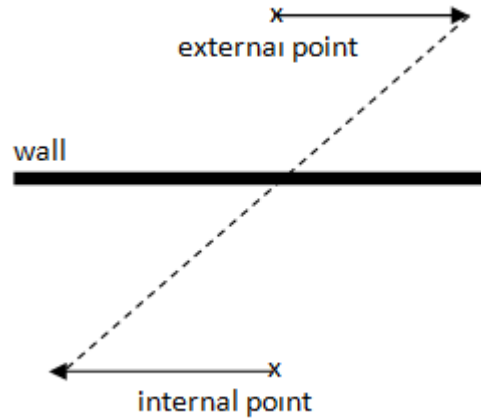
If the boundary points conform with grid points the forcing terms can be applied directly, otherwise smoothing can be introduced. This formulation allows for solid immersed boundaries without the severe time-step limitations associated with high stiffness and is more practical than elastic interface or feedback type approaches for higher Reynolds number flows.

$$\mathbf{f}(\mathbf{x}, t) = - \left( \nabla p - \mu \nabla^2 \mathbf{u} + \nabla \cdot \mathbf{u}\mathbf{u} + \frac{(\mathbf{U} - \mathbf{u})}{\Delta t} \right) (\mathbf{x}, t) \text{ for } \mathbf{x} = \mathbf{X} \quad (7.33)$$

Where  $p$   $\mu$  are the pressure and the fluid's viscosity and density and  $u$  and  $U$  are the velocity and the velocity at the boundary to be prescribed.

Mohd Yusof [55] assigned immersed no-slip walls using a technique called mirroring, whereby the velocity at a point interior to as surface is forced using Equation 7.33 such that when the interior and exterior point velocities are interpolated to the boundary surface the velocity is zero. This is shown in Figure 7.2.

Uhlman[56] later incorporated Peskin's regularised delta function into the method using a finite-difference fractional-step solver. This resulted in a robust approach free of the strong timestep restrictions of continuous forcing approaches. Pinelli *et al.* [57] improved on this method further by introducing a Reproducing Kernel Particle Method to define spreading and interpolating kernels which conserved force and moment integrals regardless of the underlying grid.



**Figure 7.2.:** Mirroring method for applying no slip boundary conditions

Chern *et al.* [47] used a discrete-forcing immersed boundary method in a finite volume framework to investigate the evolution of oscillatory flow in arrays of cylinders.

Anupindi *et al.* [58] use a mirroring/reconstruction approach to simulate turbulent flow in an aortic aneurysm using large eddy simulation. Often with immersed boundary solvers much of the domain is exterior to flow domain leading to computational overhead. To address this a multiblock methodology was incorporated, meaning that rather than a single cartesian mesh engulfing the immersed structure within which the solution is of interest, a number of connected smaller blocks can fill the domain of interest more efficiently, and so lead to less computationally wasteful simulations. Test cases included laminar flow over a sphere and a backward facing step, and blood flow simulations of a patient specific thoracic aortic aneurysm.

Discrete forcing approaches are effective for static geometries. They also have a significant advantage in terms of stability especially for higher Reynolds number simulations.

### 7.2.4. Ghost cells and Reconstruction techniques

The basis of ghost cell methods is described by Tseng *et. al* [59] and shares many similarities with the forcing/mirroring approach described in the previous section.

Ghost cells are cells within submerged solid domains which have at least one neighbour in the fluid domain. These ghost cells are used to enforce the boundary condition on the immersed boundary. This is achieved by assigning each ghost cell an interpolation scheme that implicitly incorporates the boundary condition.

This is an attractive technique if a solution is only required on one side of an interface. The choice of a stencil for the ghost cells is not straightforward.

Recently Lee *et al.* [60] utilised a sharp interface ghost-cell immersed boundary method. A mass source and sink algorithm is coupled to the solution to improve local conservation and control non-physical force oscillations on the boundary surface. A fully implicit time integration scheme is used to allow efficient simulations at CFL number. The capabilities of this approach are demonstrated on test cases including decaying vortices, vortex shedding behind a circular cylinder, and flow around a suddenly-moving circular cylinder, a flapping wing, an oscillating circular cylinder and an oscillating sphere.

Xia *et al.* [61] recently improved the implementation of the Dirichlet and Neumann boundary conditions to ensure consistency. A study was also undertaken using a high order ghost cell based immersed boundary method investigating the influence of different thermal boundary conditions on local Nusslet number over heated stationary sphere particles at different Reynolds numbers. Forced convection around a cluster of sphere particles was also simulated.

.

The Ghost Cell method allows for an accurate resolution of the boundary layer, although the choice of stencils integration schemes makes the implementation less straightforward than the methods previously discussed in this chapter. Ideally it is suited to solid objects or interfaces where a solution is not required on one side.

## Cut cell methods

Alternatively the constraints can be more directly introduced to the discretisation itself by cutting the cells (eg a finite volume cut-cell approach) - this requires a strategy for merging cells and avoiding stability issues arising from small cells appearing due to the boundary. Cut cell methods however conserve mass and momentum.

Meinke *et al.* [62] utilised a cut-cell approach, adapting with a smooth discrete formulation to handle the abrupt changes where multiple cells are (often simultaneously) created or destroyed as the boundary crosses the grid. A level set approach was used to locate the boundary. Test cases included flow past a transversely oscillating (forced) cylinder and flow within a simplified internal combustion engine with and without piston and valve movement.

Schneider *et al.* [?] focused on eliminating the unphysical oscillations occurring when cut-cell methods are extended to moving-boundary problems and extended the method to three dimensional viscous flows. The volume fractions of cells intersected by the boundary can be very small leading to problems in the time integration scheme. Fluid cells become cut, solid cells or small cells as the boundary moves leading to abrupt changes in



the discretization operator. The resulting unphysical oscillations were reduced by varying the discretization operators to take this into account. Validation cases included flow past a transversely oscillating circular cylinder, inviscid transonic flow over an oscillating NACA 0012 airfoil and flow driven by an oscillating sphere.

Gretarsson *et al.* [63] developed a conservative semi Lagrangian convective scheme capable of compensating for the small cut cell volumes which result in a loss of stability as discussed above. The robust nature of this hybrid scheme is investigated on a number of compressible flow test cases, namely constrained deforming thin shell and asymmetric shock reflection off a rigid cylinder.

Cut cell methods are challenging to implement as avoiding and treating small cells, merging cells and filling recently cleared cells requires a robust and complex algorithm. The cut cell method is however strongly conservative and provides a sharp conformal boundary onto which the boundary conditions can be directly assigned.

## 7.3. The immersed interface method

In the immersed interface method the discretisation of the operators  $\nabla$ ,  $\nabla \cdot$  and  $\nabla^2$  in the Navier Stokes equations are altered in cells local to the interface and explicit terms introduced to capture the jump conditions across the interface without the need for smoothing. This allows for a second order accuracy to be reached in the vicinity of the boundary. This technique was originally developed by Leveque and Li [64], removing the smoothing of the boundary forces, and so the associated errors.

Brehm *et al.* [65] improve on the Immersed Interface Method by the addition of a prior

check of the numerical stability. Their method allowed for irregular finite-difference stencils to be isolated from the rest of the computational domain, the introduction of a local stability constraint (employed along side the local Taylor-series expansion) and constraints to be determined on the design of these irregular finite-difference stencils. Investigations were undertaken of pulsatile stenotic flows.

More detailed overviews of the immersed interface Borazjani *et al.* [50] use the Curvilinear immersed boundary method, incorporating overset-curvilinear grids to handle multi-connected geometries undergoing arbitrarily large deformations. Curvilinear grids allow for efficient refinement near boundaries. The work focuses on ensuring globally conservative interpolation at grid interfaces suitable for incompressible flow fractional step method. The method is verified and validated against experimental data, and its capabilities are demonstrated by simulating the flow past multiple aquatic swimmers and the systolic flow in an anatomic left ventricle with a mechanical heart valve implanted in the aortic position.ethod for a variety of partial differential equations, including the Navier-Stokes equations, can be found in Li and Ito [66] and in the works of Le *et al.* ([?, 43, ?, 42])

Lagrangian multiplier type methods such as that of [?, 67] solve problems using immersed boundary methods which remove both divergence and slip on the boundary in the projection step. This approach is very useful for applying immersed boundary conditions using fixed boundaries, although it becomes less effective when the boundary moves as the Poisson equation structure changes with the interface position, meaning Fast Fourier transform or Cholesky methods cannot be used to find a solution.

## Other approaches

Bigot *et al.* [68] considered stratified density flows involving fixed or moving objects using an immersed-boundary method. The solid objects are solved using Newton's equations and defined by a continuous solid volume fraction.

The flow field near boundaries was found to be correctly captured. The full set of governing equations is then used to investigate some fundamental aspects of solid–fluid interaction, including fixed and moving objects in constant and stratified-density flows. Test cases include a NACA0012 airfoil, oscillating sphere in fluid at rest and oscillating sphere in a rotating stratified fluid and flow past a sphere dragged vertically in a stratified fluid.

Uddin *et al.* [69] recently presented an immersed boundary methodology/embedded geometry approach, to solve the compressible Navier-Stokes equations with a smooth reconstruction of pressure and viscous stresses (without jumps/discontinuities in the derivatives) utilising a level set approach and high order adaptive discretisation (using both fixed high order (for LES simulations) and adaptive order Finite Difference Cartesian compressible flow solvers)

### 7.3.1. Lattice Boltzman - Immersed Boundary method

Recently many studies have targeted utilising the Lattice Boltzman method coupled with the immersed boundary method rather than using finite volume or finite difference approaches.

The coupling of Lattice Boltzman fluid simulations with the immersed boundary method can be credited to Feng *et al.*[70]. At the time the conventional IB-LBM computed the

force density explicitly and in advance. Wu *et al.* [71] incorporated an implicit correction, allowing the no-slip boundary condition to be enforced at the boundary.

Suzuki *et al.* [72] used Lattice-Boltzmann method combined with a higher-order immersed boundary method applied only near the boundary. Conventionally there is a discontinuity of the velocity gradient introduced by the treatment of the no-slip condition, making the solution only first order in space. This is avoided using a higher-order immersed boundary method for smoothly expanding the velocity field into the body domain across the boundary. Validation cases included concentric cylinders, an oscillating circular cylinder and sedimentation of a sphere, finding an improvement on first-order spatial accuracy.

Favier *et al.* [73] note that using the Lattice Boltzmann method to advance the solution eliminates the need for a pressure correction step, which is a resource intensive requirement of finite-volume and finite difference solutions, as the velocity field is directly produced. The immersed boundary method coupled to this solver uses the method of Uhlmann [56] Pinelli *et al.* [57], which do not rely on empirical parameters.

In this study, the boundary's hydrodynamic thickness is introduced precisely, rather than being a consequence of the boundary discretisation. Test cases include impulsively started plate, particle sedimentation and the flow induced beating of a single filament and a pair of filaments.

### 7.4. Conclusions

Studies such as those by Borazjani *et al.* [50], Anupindi *et al.*[58], Brehm *et al.* [65] and Watton *et al.* [74] show that immersed boundary methods are well suited to be applied to arterial/aneurysmal flows with similar accuracy to that of moving mesh methods, although the efficiency, benefit and applicability require further investigation. For this reason continuous and discrete forcing immersed boundary methods have been selected for further analysis.

## 8. Solver design

During this work a number of solvers have been developed using a range of different strategies methods and schemes. This section outlines the theory and and algorithms used.

## 9. Baseline numerical scheme.

A baseline immersed boundary method solver was developed following the formally second-order approach of Lai and Peskin [3].

This fractional step approach advances the equations of motion in two stages. A preliminary step is taken to calculate the interface position and velocities at time  $t^{n+\frac{1}{2}}$ .

The step consists of the discretisation and solution of the following continuous equations.

Firstly the boundary forces are calculated using the boundary's constitutive equation.

$$\mathbf{F}^n = S(\mathbf{X}^n) \tag{9.1}$$

This force distribution is defined on the Lagrangian grid - i.e. at the Lagrangian marker points. In order to transfer to the Eulerian grid on which the Navier Stokes Equations are progressed the force must be integrated over the boundary and projected onto the Eulerian

force field  $f^n$ .

$$\mathbf{f}^n = \int_{\Gamma} \mathbf{F}^n \delta(\mathbf{x} - \mathbf{X}^n) ds \quad (= \mathbf{F}^n H) \quad (9.2)$$

The bracketed term on the right shows the operation in terms of the discretised operator  $H$ .

The momentum equations are now advanced to time  $t^{n+\frac{1}{2}}$  with forwards Euler differencing for the convective terms  $(\rho \nabla \cdot \mathbf{u}^n \mathbf{u}^n)$  and boundary force  $\mathbf{f}^n$ , and backwards Euler differencing  $(\mu \nabla^2 \mathbf{u}^{n+\frac{1}{2}})$  for the viscous terms.

$$\mathbf{u}^{n+\frac{1}{2}} = \mathbf{u}^n + \frac{\Delta t}{2} \left( -\nabla p^{n-\frac{1}{2}} - (\nabla \cdot \mathbf{u}^n \mathbf{u}^n) + (\mu \nabla^2 \mathbf{u}^{n+\frac{1}{2}}) + \mathbf{f}^n \right) \quad (9.3)$$

As discussed in sec. 10.4 there is no independent equation for pressure. The pressure only contributes to the pressure gradient terms in the momentum equations. To solve we must use the continuity equation as a constraint.

$$\nabla \cdot \mathbf{u}^{n+\frac{1}{2}} = 0$$

In Chorin's original projection method, the solution of the momentum equations is split up into two steps. Firstly an intermediate velocity is calculated advancing the momentum



equation without the pressure gradient term.

$$\mathbf{u}^* = \mathbf{u}^n + \frac{\Delta t}{2} \left( -(\nabla \cdot \mathbf{u}^n \mathbf{u}^n) + (\mu \nabla^2 \mathbf{u}^*) + \mathbf{f}^n \right)$$

Collecting the implicit terms on the left hand side

$$\mathbf{u}^* - \frac{\mu \Delta t}{2} (\mu \nabla \mathbf{u}^*) = \mathbf{u}^n + \frac{\Delta t}{2} \left( -(\nabla \cdot \mathbf{u}^n \mathbf{u}^n) + \mathbf{f}^n \right) = RHS \quad (9.4)$$

Due to the implicit treatment of the viscous term in the calculation of this preliminary velocity, a solution to this linear system of equations must be determined. The approaches used in this work are outlined in chapter 10.

In general the intermediate velocity field will not be divergence free and so at this point the continuity equation will not be satisfied.

In the second step 9.6 is solved, adding the pressure gradient term, which is defined at time  $t^{n+\frac{1}{2}}$  for which we do not currently have a solution . This pressure gradient term is defined such that the divergence in the velocity field is removed satisfying the continuity equation 9.5. This process, in which the correction term  $-\Delta t \nabla p^{n+1}$  is added to the intermediate velocity field  $u^*$  projects the field onto a divergence free space whilst conserving

the vorticity. Methods following this rationale are known as Projection Methods.

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad (9.5)$$

$$\mathbf{u}^{n+\frac{1}{2}} = \mathbf{u}^* - \frac{\Delta t}{2} \left( \nabla p^{n+\frac{1}{2}} \right) \quad (9.6)$$

A Poisson equation for pressure (Equation 9.7) is found by taking the divergence of the semi discretised momentum equation Equation 9.6.

$$\frac{1}{2} \Delta t \nabla^2 p^{n+\frac{1}{2}} = \nabla \cdot \mathbf{u}^* \quad (9.7)$$

With the solution of Equation 9.6 and Equation 9.7 the pressure and velocities at time  $t^{n+\frac{1}{2}}$  are known.

The preliminary step concludes with the calculation the boundary's Lagrangian properties - the forces, velocities and an updated position.

Following from [chapter diredelta] the velocity is calculated by a convolution operation, integrating the Eulerian velocity field over the support of the boundary using Equation 9.8.

The boundary velocities are used to advance the boundary position using Equation 9.9.

$$\mathbf{U}^{n+\frac{1}{2}} = \int_{\Omega} \mathbf{u}^{n+\frac{1}{2}} \delta(\mathbf{x} - \mathbf{X}^{n+\frac{1}{2}}) d\mathbf{x} \quad \left( = \mathbf{u}^{n+\frac{1}{2}} H^T \right) \quad (9.8)$$

$$\mathbf{X}^{n+\frac{1}{2}} = \mathbf{X}^n + \mathbf{U}^{n+\frac{1}{2}} \quad (9.9)$$

Having the velocity available at time  $t^{n+\frac{1}{2}}$  allows the boundary position Equation 9.9 and forces Equation 9.10 to be centred in time with second order accuracy when the solution is integrated from  $t^n$  to  $t^{n+1}$ . In order to utilise this time centred approximation to the boundary force in the Navier Stokes equations a Eulerian representation is needed. Expressing Equation 9.1 Equation 9.2 at this time results in Equation 9.10 and Equation 9.11.

$$\mathbf{F}^{n+\frac{1}{2}} = S(\mathbf{X}^{n+\frac{1}{2}}) \quad (9.10)$$

$$\mathbf{f}^{n+\frac{1}{2}} = \int_{\Gamma} \mathbf{F}^{n+\frac{1}{2}} \delta(\mathbf{x} - \mathbf{X}^{n+\frac{1}{2}}) ds \quad \left( = \mathbf{F}^{n+\frac{1}{2}} H \right) \quad (9.11)$$

In the main step, we are using Crank Nicholson scheme [75] so all of the temporal dis-

cretisation is centred in time.

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \left( -\nabla p^{n+\frac{1}{2}} - (\rho \nabla \cdot \mathbf{u} \mathbf{u})^{n+\frac{1}{2}} + \frac{1}{2} \left( \mu \nabla^2 \mathbf{u}^n + \mu \nabla^2 \mathbf{u}^{n+1} \right) + \mathbf{f}^{n+\frac{1}{2}} \right) \quad (9.12)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad (9.13)$$

The same approach is used to advance 9.12 and 9.13 as in the substep.

Interestingly, in this method the implicit term is unchanged, meaning that the same linear solver can be used, it only remains to add the appropriate term to the explicit term (RHS).

$$\mathbf{u}^* - \frac{\Delta t}{2} \mu \nabla^2 \mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \left( -\nabla p^{n+\frac{1}{2}} - (\rho \nabla \cdot \mathbf{u} \mathbf{u})^{n+\frac{1}{2}} + \frac{1}{2} \mu \nabla^2 \mathbf{u}^n + \mathbf{f}^{n+\frac{1}{2}} \right) = RHS \quad (9.14)$$

Once solved, the final velocities and positions of the boundary points are calculated in preparation for the next time step.

$$\mathbf{U}^{n+1} = \int_{\Omega} \mathbf{u}^{n+1} \delta(\mathbf{x} - \mathbf{X}^{n+1}) d\mathbf{x} \quad (= \mathbf{u}^{n+1} H^T)$$

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \mathbf{U}^{n+1} \Delta t$$

In order to incorporate immersed boundaries into our solution operators must be constructed for the spreading and regularisation operators. These operations have been assembled in a number of different ways, however for the purposes of algorithm development it is often useful to construct the operators as sparse matrices for versatility.

The coefficients of the delta functions are calculated and stored in arrays of size  $[N_b, N_s]$ . These values to populate a sparse matrix  $\mathbf{H}$  of size  $[N_b, nm]$ .

The force spreading operation is represented by multiplying the boundary forces by the matrix  $\mathbf{H}$  scaled by the boundary spacing. The operation interpolating the Eulerian velocity fields to the Lagrangian points is multiplication of the velocity field with the transpose of  $\mathbf{H}$ .

A strategy for populating  $\mathbf{H}$  is described below.

1. Find the cell indices of each Lagrangian marker in each dimension.  $\mathbf{i} = \text{floor}(\frac{\mathbf{X}}{\Delta \mathbf{x}} + \frac{1}{2})$
2. Calculate the indices of the cells that satisfy the compact support conditions for every point  $\mathbf{i}_s = \mathbf{i} + [-N_s : N_s]$
3. Find the distances of each cell with index  $\mathbf{i}_s$  from its Lagrangian marker  $\mathbf{r} = \mathbf{X} - \mathbf{i}_s \Delta \mathbf{x}$
4. Calculate the value of  $\phi$  using a delta function expression in each direction such as Equation 7.7 or Equation 7.21.

5.  $\mathbf{H}$  is filled using indices calculated from  $\mathbf{i}_s$  with the relevant strides and the outer product of  $\phi(r_x) \otimes \phi(r_y)$

If staggered grids are used then two matrices  $\mathbf{H}_x$  and  $\mathbf{H}_y$  are created, with the Lagrangian marker positions defined relative to each grid.

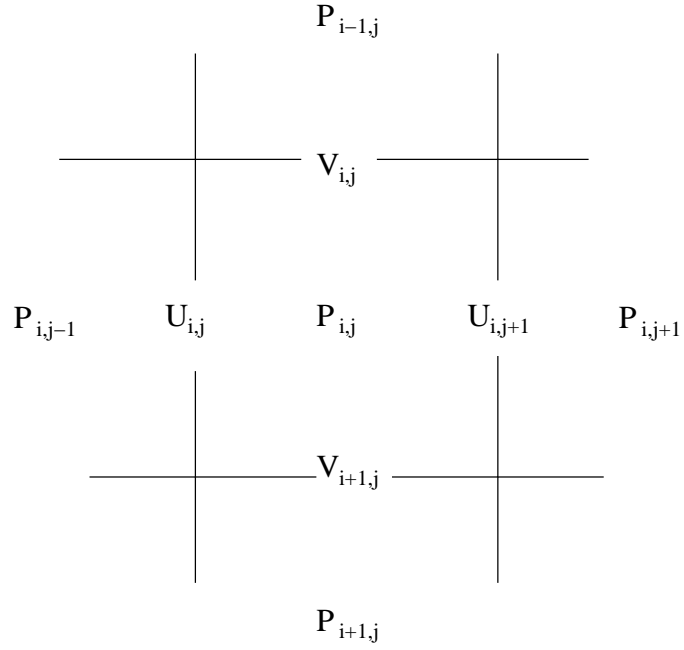
The temporal discretisation is similar to that described in chapter 8 with the additional complications of the interface, this approach matches that used by Lai *et al.* [3].

## 9.1. Discretisation in space

Colocated and staggered (MAC) finite difference grids, and a colocated finite volume discretisation have been implemented. The MAC grid is represented schematically in Figure 9.1.

On both grids the discretisation of the operators in Equation 10.9 - Equation 10.11 is handled using centred schemes. In the case of the MAC grid, expressions the gradients of the pressure are available at pressure cell faces, which is conveniently where the velocities are stored, and the gradients of the velocities are available at the cell centres where the pressure is stored. This results in a strong coupling between pressure and velocity. Further details of grid selection can be found in Ferziger et al. [76].

On equally spaced grids finite difference operations are straightforward to implement. In each direction the centred operations on the colocated grid are defined by Equation 9.15 and Equation 9.16 below.



**Figure 9.1.:** Positioning of the velocity and pressure components on the MAC grid

$$\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i \approx L_x = \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{(\Delta x)^2} \quad (9.15)$$

$$\left(\frac{\partial \phi}{\partial x}\right)_i \approx D_x = \frac{\phi_{i+1} - \phi_{i-1}}{\Delta x} \quad (9.16)$$

Using the standard centred approximation for the differencing terms we arrive at the following numerical discretisation for the one dimensional operator  $L$  with different boundary conditions.

On colocated finite difference grids the skew symmetric differencing is often used ([38]) for the convective terms

	<i>Neumann</i>		<i>Dirichlet</i>
<i>Colocated</i>	$\begin{bmatrix} -2 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix}$		$\begin{bmatrix} -2 & 2 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 2 & -2 \end{bmatrix}$
<i>Staggered</i>	$\begin{bmatrix} -3 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -3 \end{bmatrix}$		$\begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$
	<i>Periodic</i>		
	$\begin{bmatrix} -2 & 1 & 0 & 0 & 1 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix}$		

**Table 9.1.:** Matrix representation of the Laplacian for various boundary conditions

$$S(\mathbf{u}_n)\phi = \frac{\mathbf{u}_n D_n(\phi) + D_n(\mathbf{u}_n \phi)}{2}$$

It is straightforward to construct a sparse matrix in two dimensions with which to solve Poisson or Helmholtz type equations using a Kronecker product Equation 9.17 of the one dimensional differencing operations with the identity matrix. For these grids the boundary conditions incorporated into the one dimensional approximation are sufficient



for the multidimensional problem.

$$\Delta^2 = L = D_x \otimes I + I \otimes D_y \quad (9.17)$$

For staggered grids Equation 9.17 returns the five point Laplacian. Using a discretised Laplacian constructed using Equation 9.17 leads to the discretisation operations used in the momentum and pressure equation being consistent. This results in the projection method used to remove the divergence in the velocity fields being exact.

On centred grids the five point Laplacian and centred schemes can still be used in the discretisation of the momentum equations, however if it is used in the Poisson equation for the pressure, the projection method will no longer be exact. This results in an approximate projection method [77].

Alternatively the Laplacian operator in the Poisson equation can be constructed using Equation 9.17 with centred differencing operators on the colocated grid. In this case the Laplacian stencil samples points at double spacing this reduces the accuracy of the pressure, but does not strongly influence the velocity fields for reasons explained earlier in section 7.1.

## 10. Linear Solvers

Immersed boundary approaches allow for significant simplifications to be made to the fluid solver. The simplest approach is to solve the Navier-Stokes equations on uniformly spaced Cartesian grids with periodic boundary conditions and constant discretisation coefficients over the domain. With this simplification the pressure and momentum equations can be solved using either Fast Fourier methods or fast Poisson/Helmholtz solvers. Although external boundary conditions are useful in many potential cases, they can be replaced with immersed boundaries.

The limitation to constant coefficients over the domain allows for no implicit or semi implicit treatment of the convective term in the momentum equations. Rhie-Chow interpolation which is used for pressure-velocity coupling on colocated grids also requires the coefficients of the Poisson equation to vary over the domain. This means that with the above assumptions, we must either use a staggered (eg MAC) grid, or settle for a lower order approximation for pressure, which is a vital parameter balancing the boundary forces.

A minimal strategy to advance the Navier-Stokes could consist of explicit integration of the momentum equations in time, calculating the viscous and unsteady terms using values from the previous time-step, but would still require the solution of a Poisson equation for

pressure. Therefore the minimum required from a linear solver is the solution of a Poisson equation with periodic boundary conditions on a equally spaced Cartesian grid.

### 10.1. Spectral approach

When the problem is discretised on a uniform orthogonal Cartesian grid with periodic odd or even boundary conditions and constant viscosity we can solve efficiently in Fourier space in the order of  $N \log(N)$  operations. The solution of the Poisson's equation can be found via Eigenfunction expansion [1]. The discrete Fourier transform and its inverse transform are defined

$$\begin{aligned}\phi_{ij} &= \frac{1}{JL} \sum_{m=0}^{J-1} \sum_{n=0}^{L-1} \hat{\phi}_{mn} e^{-2\pi i j m/J} e^{-2\pi i n l/L} \\ \hat{\phi}_{mn} &= \sum_{j=0}^{J-1} \sum_{l=0}^{L-1} \phi_{jl} e^{2\pi i m j/J} e^{-2\pi i n l/L}\end{aligned}\tag{10.1}$$

The spacial grid points of the Cartesian mesh to are defined by  $i$  and  $j$  and the spatial frequency domain is defined by the indices  $m$  and  $n$ . Both of these indices vary between 0 and the number of mesh points in each direction (M or N) -1 for the periodic case. In Fourier space it is straightforward to shift the spatial field values an integer number of grid points in orthogonal directions and thus calculate the eigenvectors required to find a solution in the manner outlined below.

To solve the Poisson equation the right hand side of the equation is first transformed into

the Fourier domain using Equation 10.1 and a solution is found using Equation 10.2.

$$(\lambda I - A)\hat{u} = \hat{y} \quad (10.2)$$

where  $\hat{y}$  is the transformed right hand side of the equation (the explicitly calculated terms).

An expression for the eigenvector can be found by summing the expressions at the relevant grid points, i.e. at offsets  $i = -1, 0, 1$  and  $j = -1, 0, 1$  using shift theorem.

With the use of trigonometric identities this result simplifies to the following expression in two dimensions for the periodic case.

$$\lambda = 2 \left( \cos \frac{2\pi m}{J} + \cos \frac{2\pi n}{L} - 2 \right)$$

This expression is simply the sum of the one dimensional eigenvectors.

Alternative boundary conditions can be modelled using the discrete sine and cosine transforms.

For homogenous Dirichlet boundary conditions the discrete sine transform can be used. In this case, for gridpoints in which the stencil extends beyond the grid, i.e. where cells are bordering an external boundary, the dataset is extrapolated such that a boundary value of zero is assigned for a boundary either at single (DST-1) or half spacing (DST-2) outwards from the first and final gridpoints.

The discrete cosine transform extends the dataset in a manner equivalent to assigning homogenous Neumann boundary conditions - the gradient of the value is assigned to zero

at the first and last grid point in the case of the DCT-1 and at half spacing from the first and last point in the case of the DCT-2.

The extrapolation can be seen in Figure 10.1.

The eigenvalues of the 5 point Laplacian operator for both the discrete sine and discrete cosine transform cases is of the form defined as follows

$$\lambda = 2 \left( \cos \frac{\pi m}{J} + \cos \frac{\pi n}{L} - 2 \right)$$

where the only differences are in the terms  $\frac{m}{J}$  and  $\frac{n}{L}$  i.e. it is only required to account for the differencing in the indexing in the even and odd transforms with staggered and colocated endpoints.

In one dimension these operations result in same matrix structures as displayed in Table 9.1 in the previous section.

In the case of even or periodic transforms the solution (where  $A=0$ ) requires blocking out the zero frequencies before each inverse transformation is applied.

On the MAC grid we also wish to assemble forward and backward differencing operators in order to calculate the pressure and velocity gradients on each grid at half spacing. In the periodic case the finite difference approximations to these forwards and backwards differencing approximations in the Fourier domain will have the form shown below

$$D_{xf} \hat{\phi} = \hat{\phi} \left( e^{2\pi im/J} - 1 \right)$$

$$D_{xb} \hat{\phi} = \hat{\phi} \left( 1 - e^{-2\pi im/J} \right)$$

## 10.2. Iterative approach

Iterative solution methods have been developed in order to allow for approaches without the limitation of constant coefficients. This has been done to allow, amongst other improvements, methods with alterations to the discretisation local to boundaries, and Rhie-Chow interpolation approaches for colocated finite volume solutions. Firstly Gauss and Jacobi methods are discussed then the extension to multigrid which is practical for larger cases.

### 10.2.1. Gauss and Jacobi solver

Gauss-Seidel iteration consists of the following update to an equation of the form Equation 10.3.

$$Ax = b \tag{10.3}$$

$$x^{(k+1)} = L^{-1} \left( b - Ux^{(k)} \right)$$

where L is the lower coefficient and U is the upper coefficient.

A Jacobi update is defined as

$$x^{(k+1)} = D^{-1} \left( b - Rx^{(k)} \right)$$

where  $D$  is the diagonal coefficient, and  $R$  is the remainder. Gauss and Jacobi iteration schemes are very slowly converging, requiring in the order of  $N^2$  iterations, ie the number of iterations scales with the number of mesh points. However these approaches are effective smoothers for multigrid solutions.

### 10.2.2. Multigrid Solver

Equation 10.3 can be solved with much fewer operations using a Multigrid technique. The partial differential equations must first be discretised on a series of increasingly coarse grids. With the use of equally spaced Cartesian grids this is a straightforward operation. An overview of this approach for finite difference type equations is given in Jorge *et al.* [78].

Spreading and restriction operators are required which for the case of a finite volume type solution (where coefficients can be added over the faces between levels) are displayed below.

For the finite difference discretisations bi-linear or half weighting operators for prolongation and interpolation of the form below can be used

$$R_{half} = \begin{bmatrix} 0 & \frac{1}{8} & 0 \\ \frac{1}{8} & \frac{1}{2} & \frac{1}{8} \\ 0 & \frac{1}{8} & 0 \end{bmatrix} \quad (10.4)$$

$$R_{biharmonic} = \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \quad (10.5)$$

A V-cycle algorithm was implemented whereby the following operations are performed

1. The discretised equation ( $Ax = b$ ) is relaxed on the finest grid using the Gauss update from the previous section.
2. The residual ( $r = Ax - b$ ) after relaxation on this grid is calculated and restricted onto the next coarsest grid where an equation for the correction is assembled using this error ( $Ax_1 = b_1 = Rr$ ) where  $R$  represents the restriction operation.
3. This equation is relaxed using Gauss updates and these initial steps repeated recursively until the error is found on the finest grid.
4. An upward sweep begins, the error solved on the finest grid is prolonged to the next coarsest grid, adding to the solution previously stored there ( $A_{n-1}(x_{n-1} + Px_n) = b_{n-1}$ ).
5. A number of Gauss updates are applied to relax the solution.
6. The process is repeated up to the finest grid.



7. The entire V-cycle is repeated until the desired convergence criterion is met.

The process outlined above is shown schematically in Figure 10.3.

## 10.3. Comparison

To select an appropriate approach the Laplace equation was solved on a square domain with periodic boundary conditions, using solvers written with each method.

The Laplace equation governs numerous physical problems, such as heat diffusion, electromagnetic potential or velocity potential in irrotational flow.

Table 10.1 gives the time taken for the solution using the various approaches outlined above to solve either directly or to specific residuals. To get the best performance out of the multigrid and fast Fourier solvers the number of grid points in each direction were set to powers of 2. The densest grid contains  $6.7e7$  cells in total and takes 2.245 seconds to solve using the Eigenfunction expansion method.

Solver	n=8	n=9	n=10	n=11	n=12	n=13
FFT	8.85e-4	5.11e-3	2.42e-2	0.1043	0.4514	2.245
Multigrid $r=1E-6$	1.383e-2	5.8e-2	0.245	0.956	1.43	-
Gauss $r = 1E-6$	10.995	48.559	215.05	-	-	-

**Table 10.1.:** Computational time requirement for various linear solution strategies for  $2^n \times 2^n$  grid points

Ideally a multigrid solver should be comparable or be able to outperform fast Fourier methods as they should scale with  $O(N)$  rather than  $O(N \log(N))$ .

This was not found to be the case with the current implementation as can be seen in Table 10.1. It is believed that this is due to FFTW, the numerical library used in the

based solution being exceptionally efficient in both its memory footprint parallelisation and optimisation.

Comparatively there is an extra overhead in multigrid solutions, and the solver could be improved in a number of ways, including the use of better smoothing methods such as red-black Gauss-Seidel using which further optimisations could be achieved ([78]).

### 10.4. Fractional Step methods

For compressible flows, the Navier-Stokes equations can be solved by using the continuity equation to calculate the fluid's density, the pressure can be determined using the fluid's equation of state and the velocity components can be solved using the momentum equations. For incompressible flows the solution of the Navier Stokes equations is not straightforward

This is for two main reasons (as outline in Ferziger *et al.* [76]). Firstly, there is no independent equation for pressure. The pressure only contributes to the pressure gradient terms in the momentum equations. Secondly, the continuity acts only as a kinematic constraint (as the density is constant), meaning that to solve for one velocity component the others must be known.

There are number of approaches to pressure-velocity coupling which are described and compared in this section.

The Navier Stokes equation can be written in block matrix form as follows

$$\begin{bmatrix} A & G \\ D & 0 \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} r^n \\ 0 \end{bmatrix} + \begin{bmatrix} bc \\ bc \end{bmatrix}$$

Where A is the sum of the implicit terms of the advection-diffusion part of the momentum equations, G and D are gradient and divergence operators respectively,  $r^n$  is the explicit right hand side of the momentum equations and bc are the boundary conditions. For the reasons outlined above the system is indefinite.

### Artificial compressibility methods

The most straightforward approach towards making sec. 10.4 solvable is to introduce artificial compressibility

$$\begin{bmatrix} A & G \\ D & \lambda I \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} r^n \\ 0 \end{bmatrix} + \begin{bmatrix} bc \\ bc \end{bmatrix}$$

This is an approximation with order of accuracy  $\lambda$ , which is a non-physical parameter the size of which has a large impact on the stiffness of the solution.

### Fractional Step methods

The fractional step method was first proposed by Harlow et al, and Chorin et al. [79]

At each time step an incomplete form of the momentum equations is integrated, then a correction is applied removing the divergence of the velocity field.

and Chorin

In Chorin's original projection method, the solution of the momentum equations is split up into two steps. Firstly an intermediate velocity is calculated using the momentum equation with Euler explicit advancement and without the pressure gradient term.

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -(\mathbf{u}^n \cdot \nabla) \mathbf{u}^n + \nu \nabla^2 \mathbf{u}^n \quad (10.6)$$

Then the pressure gradient term is added in the second step

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\nabla p^{n+1} \quad (10.7)$$

The right hand side of this equation requires the pressure at time  $t^{n+1}$ . Noting that the final velocity field must be divergence free, and taking the divergence of the pressure correction equation yields equation Equation 10.8

$$\nabla \cdot \mathbf{u}^{n+1} = 0$$

$$\nabla^2 p^{n+1} = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t} \quad (10.8)$$

It can be noted that the correction term added to the velocity field  $-\Delta t \nabla p^{n+1}$  projects the field  $u^*$  onto a divergence free field whilst conserving the vorticity, thus this known as a Projection method.

The conservative properties of the scheme depend strongly on the choice of discretisation method for the divergence, gradient and laplacian terms.

Kim and Moin[80] modified the original Chorin method for use on a staggered finite volume grid.

A fully discrete approach is described by Perot et al.[77]

It was noted in Perot that the Fractional Step method of Chorin outlined above is the continuous analogue of the block LU decomposition of the system

$$\begin{bmatrix} A & (\Delta t A) G \\ D & 0 \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} r^n \\ 0 \end{bmatrix} + \begin{bmatrix} bc \\ bc \end{bmatrix}$$

The additional parameter  $\Delta t A$  introduces a first order error in time

Performing block LU decomposition ([77]) yields the system

$$\begin{bmatrix} A & 0 \\ D & -\Delta t D G \end{bmatrix} \begin{bmatrix} u^* \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} r^n \\ 0 \end{bmatrix} + \begin{bmatrix} bc \\ bc \end{bmatrix}$$

$$\begin{bmatrix} I & \Delta t G \\ 0 & I \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} u^* \\ p^{n+1} \end{bmatrix}$$

The momentum equations are solved to determine the flow velocities, leaving the continuity equation to determine the pressure field. As no pressure term appears in the continuity equation, the momentum equations must be reformulated to solve for pressure using the continuity equation as a constraint.

During this work a number of approaches to solving the Navier Stokes equations have been investigated with an aim of finding the most suitable for application with immersed boundary methods. Many different strategies were used to advance the pressure and velocity fields, including pressure free and pressure increment projection methods and PISO schemes.

In each case the solutions were advanced with the convective and diffusive terms centred in time, leading (in the fluid discretisation) to second order temporal accuracy in the velocity fields.

To solve the Navier Stokes equation a fractional step approach similar to that of Perot [77] and initially described by Chorin is utilised.

A preliminary solution for the velocity is calculated by either neglecting the pressure at the current timestep or using an explicit value from the previous timestep. In the case for which the pressure is staggered in time the following expression is reached where the convective term can be extrapolated using two step Adams Bashforth approach and the

viscous term interpolated using second order Crank Nicholson.

$$\begin{bmatrix} I & \Delta t G \\ 0 & I \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} u^* \\ p^{n+1} \end{bmatrix} \quad \mathbf{u}^* = \mathbf{u}^n + \Delta t \left( -\nabla p^{n-\frac{1}{2}} - (\nabla \cdot \mathbf{u}\mathbf{u})^{n+\frac{1}{2}} + (\mu \nabla^2 \mathbf{u})^{n+\frac{1}{2}} + f^{n+\frac{1}{2}} \right) \quad (10.9)$$

Alternatively a fractional step can be taken with fully implicit treatment of the viscous term, and these intermediate values used to populate the centred terms and advance the solution. This approach is discussed in ??.

Dropping the pressure term for the preliminary velocity in Equation 10.9 results in a pressure free approach.

The divergence of the expression produces a Poisson equation for pressure of the form.

$$\nabla^2 p = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t} \quad (10.10)$$

For a colocated finite-volume an approach such as Rhie-Chow interpolation must be used to prevent decoupling of the solution. This effectively consists of multiplying the coefficients on the faces in the divergence term by the central coefficients of the momentum equations interpolated to the faces.

The velocity is now updated to incorporate the pressure gradient or an update to the pressure gradient depending on the formulation. For the colocated finite volume approach

these will once again be weighted by the inverse of the central terms of the momentum equation if Rhie-Chow interpolation is used.

$$\mathbf{u}^{n+1} = \mathbf{u}^{**} - \Delta t \nabla p$$

If required, the pressure can now be updated using an expression similar to the one below

$$p^{n+\frac{1}{2}} = p^{n-\frac{1}{2}} + p^* \tag{10.11}$$

Depending on the timestep and method iterations may be needed, especially if the non-linearity in the momentum equations is to be better taken into account (if the convective flux is discretised using implicit velocity and explicit momentum flux as is common in the finite volume treatment).

The arrival at the Poisson equation in the fractional step method can be visualized using three equivalent approaches, discretionary and splitting of the continuous equations in time, the LU decomposition of the discretised Navier Stokes equations – this allows for the correct treatment of boundary conditions and higher order time integration ([77]), and by considering the pressure to act as a Lagrangian multiplier. The treatment of pressure as a Lagrangian multiplier will conceptually allow the introduction of additional constraints to the momentum equations.



### 10.4.1. Timestep limitations

The explicit treatment of the non-linear convective terms introduces the Courant–Friedrichs–Lewy condition.

This is best understood by considering a one dimensional transport equation using upwind differencing for the the convective term and forward Euler for the unsteady term. From Ferziger et al. [76]

$$\phi_i^{n+1} = \phi_i^n + \left[ -u \frac{\phi_i^n - \phi_{i-1}^n}{\Delta x} + \frac{\Gamma}{\rho} \frac{\phi_{i+1}^n - 2 * \phi_i^n + \phi_{i-1}^n}{\Delta x^2} \right] \Delta t$$

This expression simplifies to

$$\phi_i^{n+1} = (1 - 2d - c) \phi_i^n + d \phi_{i+1}^n + (d + c) \phi_{i-1}^n$$

where  $d = \frac{\Gamma \Delta t}{\rho (\Delta x)^2}$  and  $c = \frac{u \Delta t}{\Delta x}$

A negative coefficient  $(1 - 2d - c)$  will result in an instability, leading us to the following definition

$$\Delta t < \frac{1}{\frac{2\Gamma}{\rho \Delta x^2} + \frac{u}{\Delta x}}$$

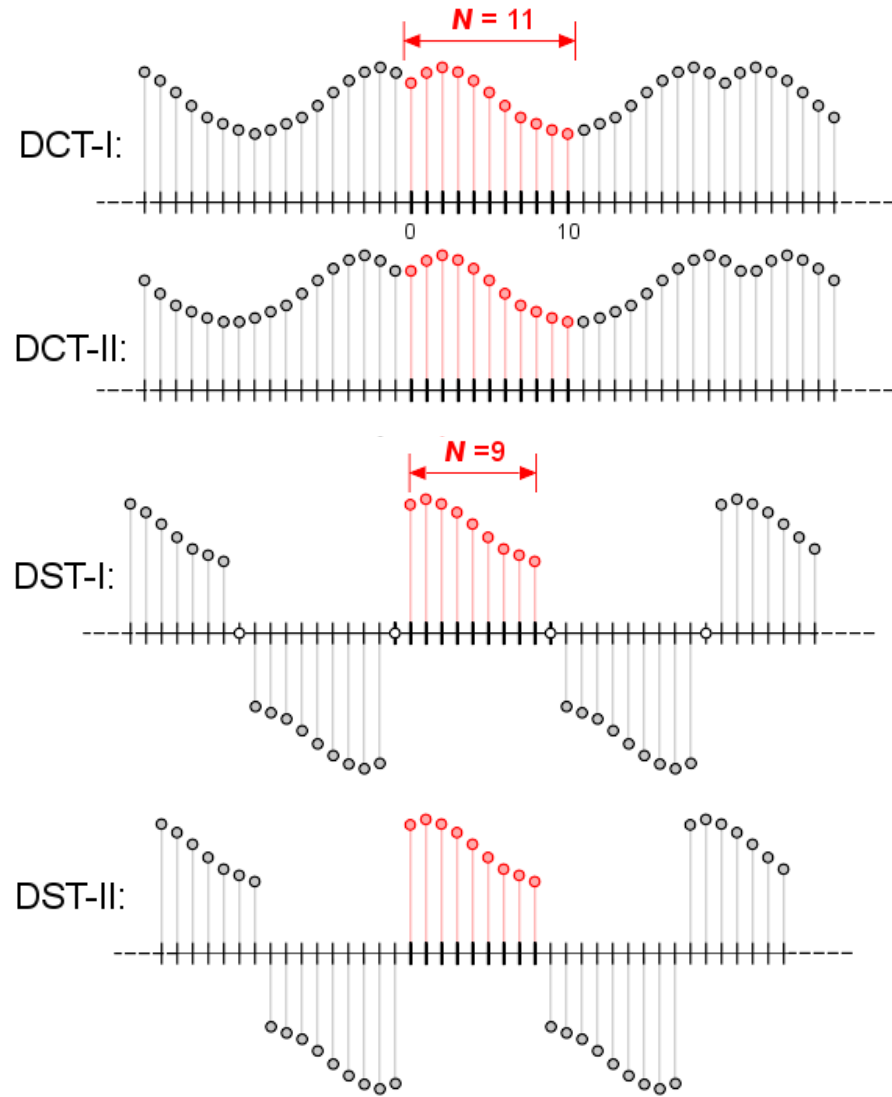
So in the cases of negligible convection the Courant number needs to be less than unity

$$C = \frac{u\Delta t}{dx} < 1$$

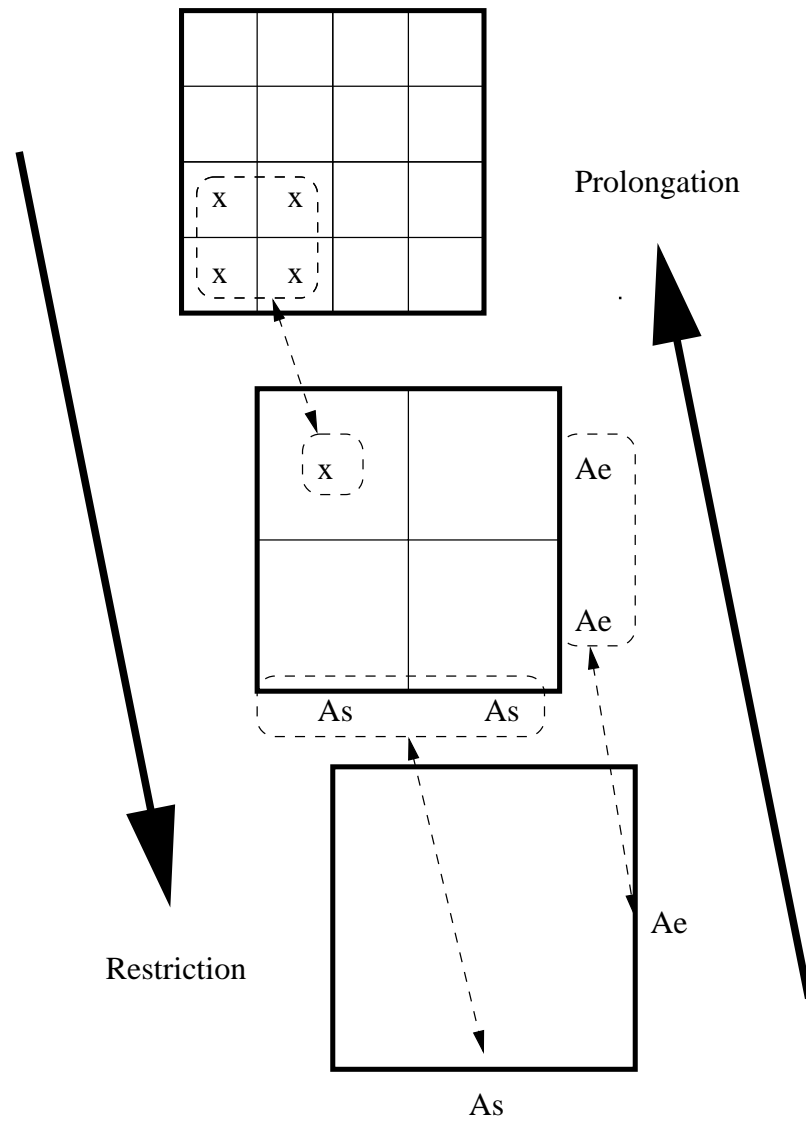
For immersed elastic fibres another limit is imposed on the timestep. A detailed analysis is provided by Stockie *et al.* [37, 81], an approximate relationship is derived by Lai *et al.* [3]

$$\Delta t \approx C \sqrt{\frac{h}{k}}$$

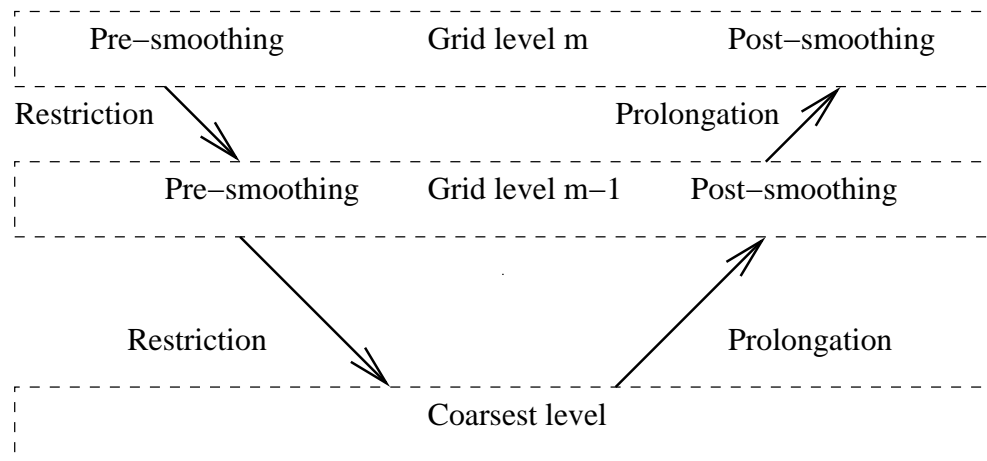
where  $h$  is the spacing of the Eulerian mesh and  $k$  is the spring stiffness.



**Figure 10.1.:** Discrete transform types used in the Eigenfunction expansion solution method [1]



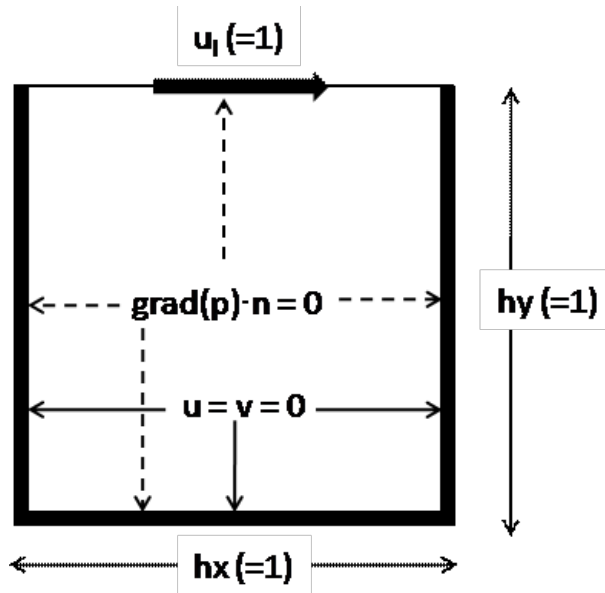
**Figure 10.2.:** Various key concepts of the multigrid strategy including the restriction and prolongation operations



**Figure 10.3.:** The V-cycle Multigrid algorithm

# 11. Fluid solver validation

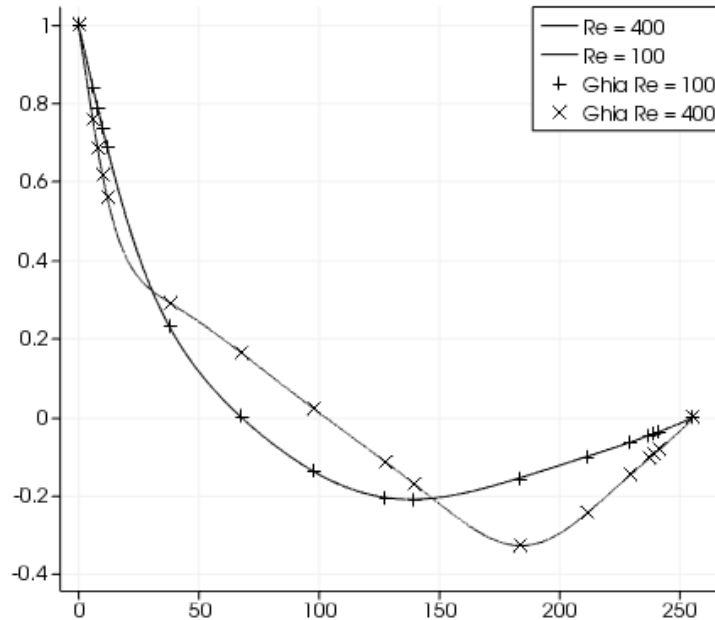
The MAC collocated fluid solver using the Eigenvalue expansion linear solver (other linear solvers produced identical results to the assigned tolerances) has been validated against the numerical results of Ghia *et al.* [2] with Dirichlet boundary conditions prescribed for the fluid velocity. The geometry and boundary conditions used are given in Figure 11.1.



**Figure 11.1.:** Boundary conditions for fluid solver validation case

The total number of cells in each direction in both cases was 256. The fields were ad-

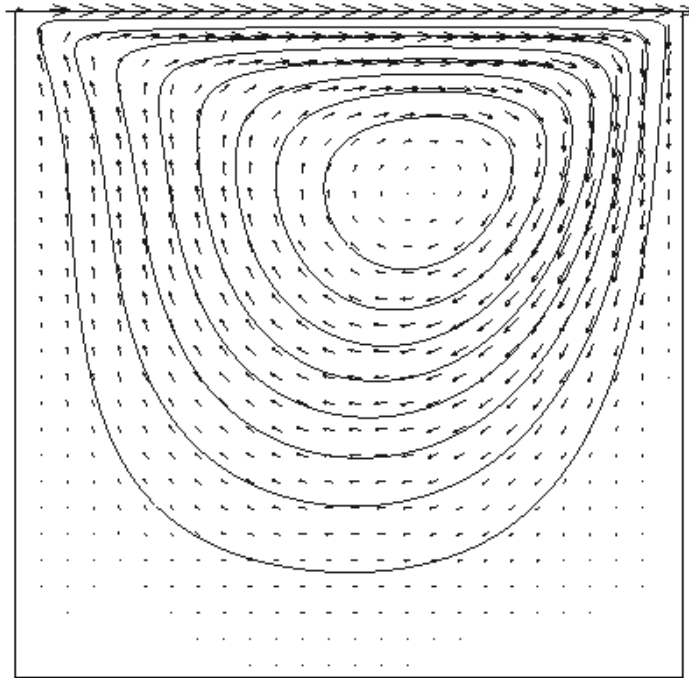
vanced to the steady state solution using the time-step defined by the CFL number being equal to unity and run until steady state solutions were reached.



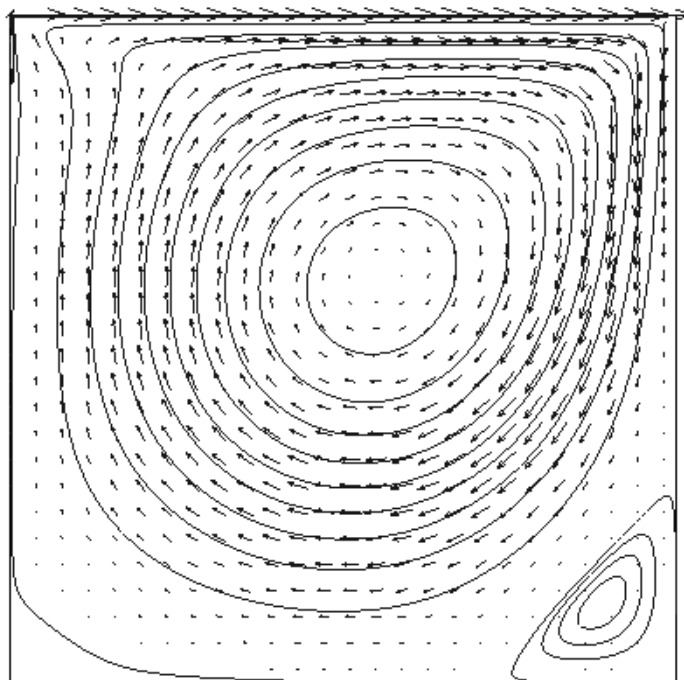
**Figure 11.2.:** Centreline x-velocity plot for the fluid solver validation case - comparison with Ghia *et al.* [2]

Figure 11.2 shows the velocities in the x-direction at  $y = 0.5$  in comparison to those generated by Ghia *et al.* [2]. Excellent agreement is found at both  $Re = 100$  and  $Re = 400$  with second order central differencing.

Figure 11.3 and Figure 11.4 show the normalized velocity vectors superimposed on the velocity magnitude contours. At  $Re = 400$ , the corner vortices can be clearly seen which will expand further as the Reynolds number is increased.



**Figure 11.3.:** Velocity vectors and streamlines for driven cavity flow with  $Re=100$



**Figure 11.4.:** Velocity vectors and streamlines for driven cavity flow with  $Re=400$



## 12. Processing complex boundaries

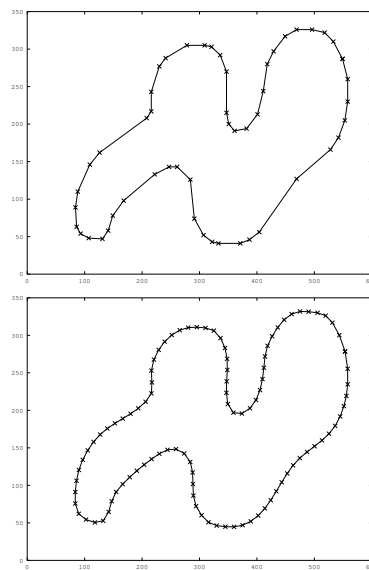
Routines have been written using the image processing library OpenCV in order to extract 2 dimensional geometries from images. The process consists of the following steps

1. An image is loaded (Figure 12.1)
2. The image is smoothed using a user defined number of smoothing steps
3. The image is converted into a binary matrix
4. Contours are extracted from the image using the Canny Edge detecting algorithm
5. Boundaries are extracted and stored in XML format

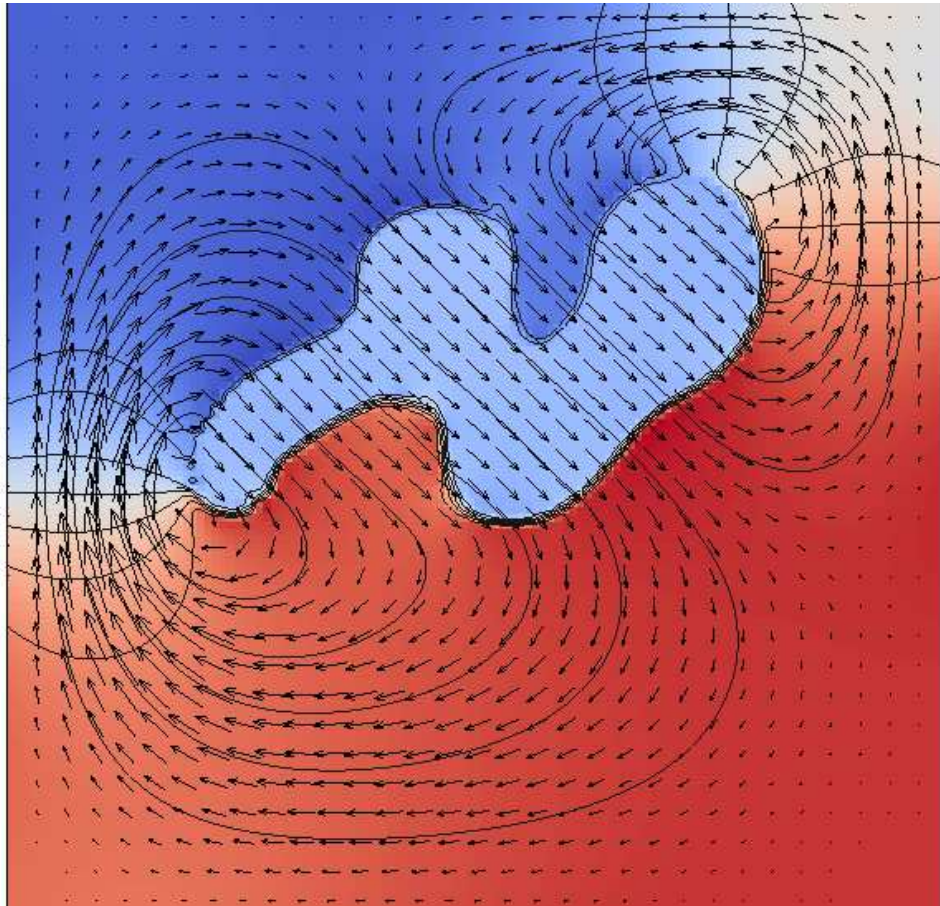
The edge extraction results in jagged edges due to the binary image sampling. This is overcome by interpolating the boundary with cubic splines fitted using a least squares approach which also allows for the sampling to be evenly spaced (see Figure 12.2 for an example). These sampled points represent the control points of the cubic spline in the solver. Sample results for a boundary velocity defined as  $(1,1)$  in a closed cavity are displayed in Figure 12.3.



**Figure 12.1.:** Sample raster image for processing



**Figure 12.2.:** Point extraction and smoothing/resampling



**Figure 12.3.:** Sample velocity vectors with overlaid pressure and streamfunction contours for arbitrary geometry with boundary velocity =  $(1, 1)$  in a cavity ( $Re=1$ )

## 13. Validation and test cases

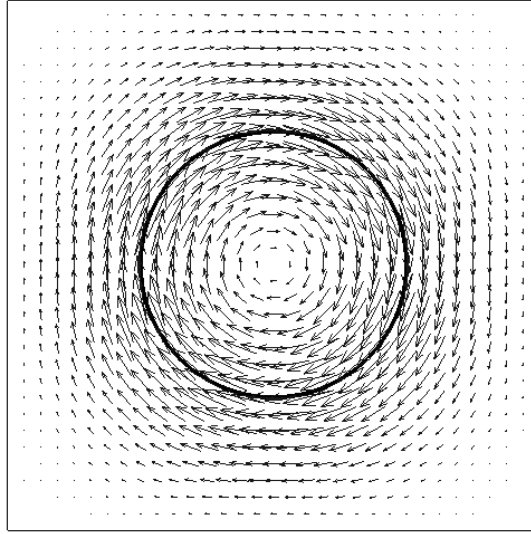
### 13.1. Driven rotational flow in a cavity

The method described in chapter 9 with the forcing from the virtual boundary approach outlined in subsection 7.2.2 has been applied to a cavity of unit length in both directions with homogeneous Dirichlet boundary conditions imposed on the velocity at the walls containing a circular immersed interface of diameter 0.5.

Forcing is calculated to drive the interface velocity tangential to the interface to 1 using Equation 7.32.

The resulting flow contained within the immersed boundary is purely rotational as can be seen in the velocity vectors displayed in Figure 13.1 and the contours of streamfunction displayed in Figure 13.2, which closely fit the boundary interface reflecting that the velocity is constant on the interface as prescribed indirectly using Equation 7.32.

The flow field internal to the immersed interface is no longer influenced by the external boundary conditions as can be seen in the plots of x-velocity across the centreline of the domain and diagonally across the domain.

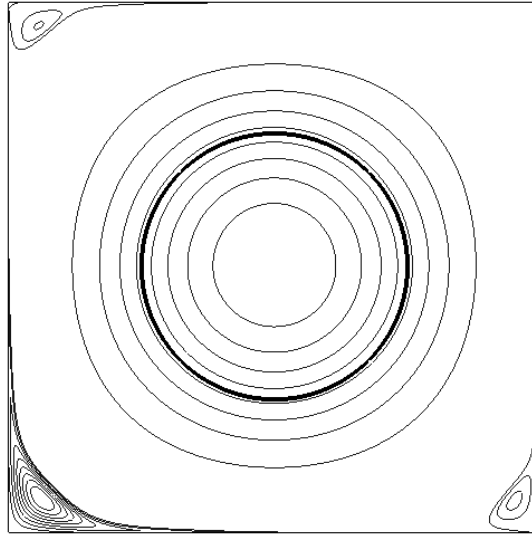


**Figure 13.1.:** Velocity vectors for rotational flow in a cavity driven by a circular immersed interface

The centreline velocity plots show, as should be expected, that the velocity increases linearly from  $-1$  to  $1$  between opposing sides of the immersed boundary, and the  $x$ -component of the velocity increases linearly from  $-\sqrt{\frac{1}{2}}$  to  $\sqrt{\frac{1}{2}}$  diagonally across the area enclosed by the immersed boundary reflecting that the magnitude of the velocity on the interface is unity.

## 13.2. Taylor Couette flow

The case of Taylor Couette flow, the fluid flow between two counter-rotating cylinders, is investigated using an identical geometry to that used in section 13.1 with an additional immersed interface with a diameter of 0.35 within the first.

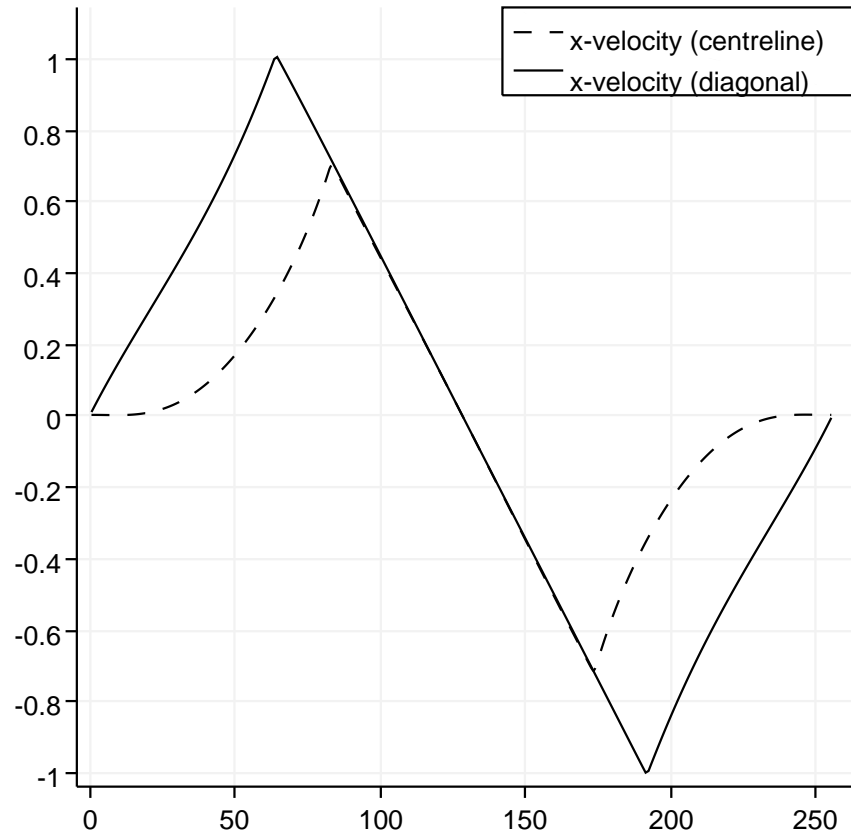


**Figure 13.2.:** Contours of streamfunction for rotational flow in a cavity driven by a circular immersed interface

In this case the forcing function on the internal interface is prescribed to set the velocity on (and resultingly inside) the inner boundary to zero once again using Equation 7.32.

As can be seen in Figure 13.4 and Figure 13.5 the contours of streamfunction once again mimic the interface, showing that the velocity at the interface is approximately constant. The velocity vectors generated are purely rotational and increasing from zero to unity between the internal and external immersed interface (Figure 13.6) as expected.

Outside of the domain inertial effects can be seen with areas of stagnation and recirculation that are dependent on the Reynolds number of the simulation.

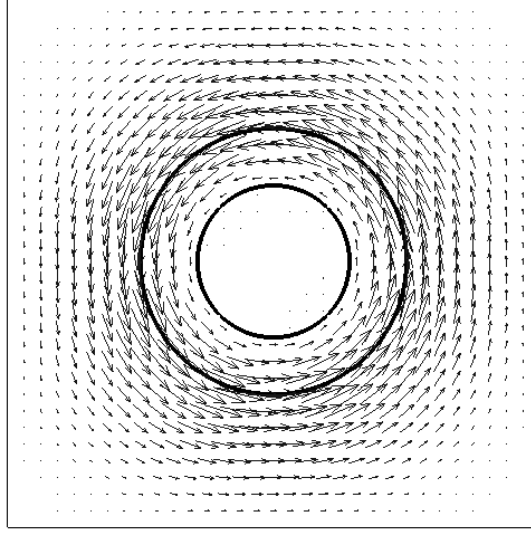


**Figure 13.3.:** Centreline and diagonally sampled x-velocity across the domain and intersecting the immersed geometry for rotational flow in a cavity generated using immersed boundaries

### 13.3. Perturbed droplet in stationary flow

In this section results are presented for the response of a perturbed droplet submerged in a viscous fluid using both the standard immersed boundary method approach and solving implicitly and semi implicitly using the BFGS method (with results discussed in section 14.1).

A boundary was initialized in the fluid domain in the shape of an ellipse, the geometry



**Figure 13.4.:** Velocity vectors for Taylor-Couette flow in a cavity driven using immersed boundaries

and boundary conditions are displayed in Figure 13.7.

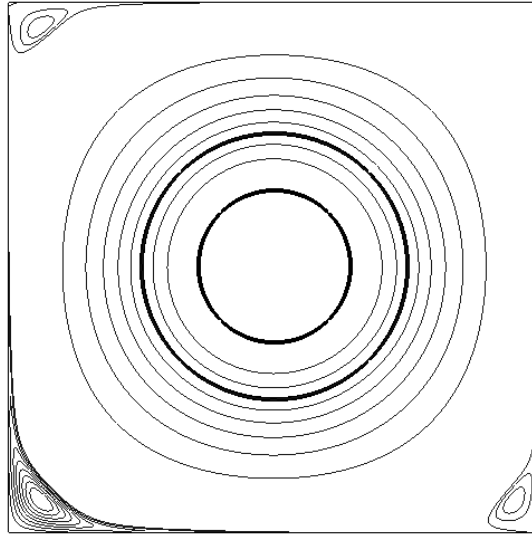
The boundary forces consist of surface tension which is modelled by the discretisation of Equation 13.1 over the length of the immersed boundary

$$F = \gamma \frac{\partial^2}{\partial s^2} X \quad (13.1)$$

Where  $\gamma$  is the surface tension.

For this study, the fluid viscosity fluid density and and surface tension were set 0.001, 1 and 1 respectively. Solutions were found using 5 different grid spacings defined in Table 13.1.



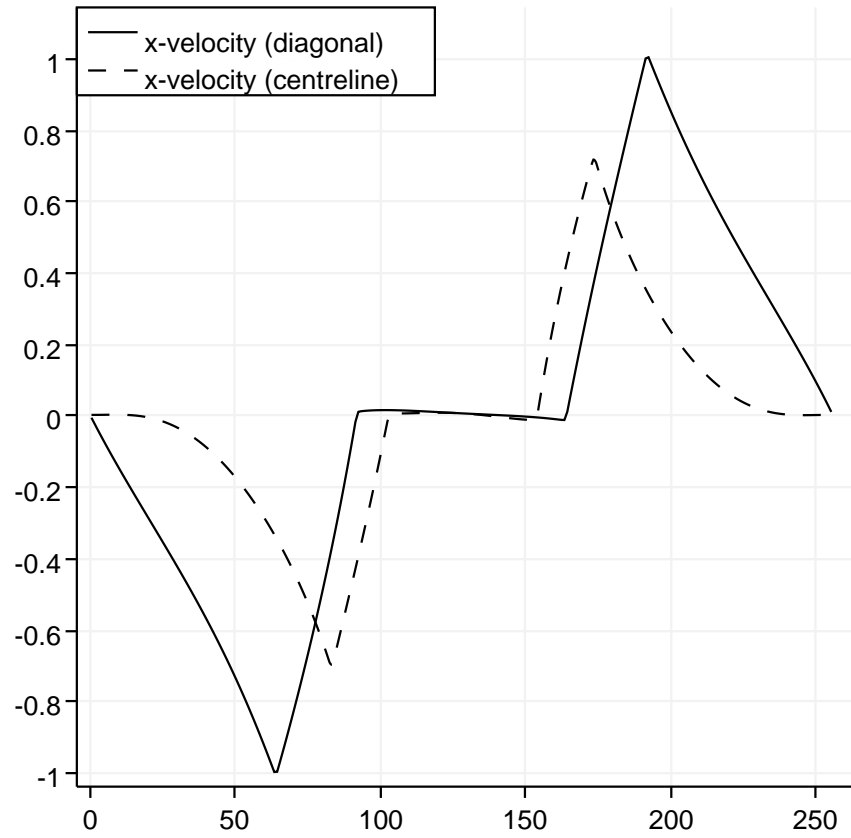


**Figure 13.5.:** Contours of streamfunction for Taylor-Couette flow in a cavity driven using immersed boundaries

This ellipse configuration is unstable, and the boundary quickly oscillates around, and after a short time to, the shape of a circle with a volume matching that of the original ellipse. The response of the droplet can be seen in Figure 13.8 where the interface position and velocity vectors are plotted at intervals of 0.15 seconds, half the period of oscillation of the droplet.

MAC and colocated grid discretisations were used and their results compared. Colocated checkerboard type solvers are often used in immersed boundary method simulations.

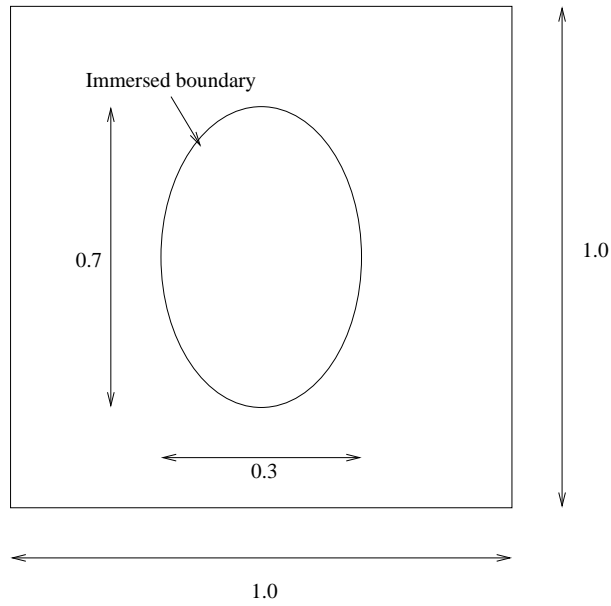
The major and minor axes of the droplet are displayed for the MAC grid case in Figure 13.10 and the colocated grid discretisation case in Figure 13.9. It can be seen that the major and minor axes of the droplet in the colocated case are shrinking significantly throughout the simulation to the point where on the coarsest grid the droplet area reduced by 20%.



**Figure 13.6.:** Centreline and diagonally sampled x-velocity across the domain and intersecting the immersed geometry for Taylor-Couette flow in a cavity driven using immersed boundaries

Volume losses were much smaller (although still significant) on the MAC grid.

The area of the droplet was calculated using the maximum and minimum extent of the droplet during the simulation and assuming the interface to be elliptical. The result is displayed in Figure 13.11. The initial oscillations occur as the droplet as the interface is not truly elliptical during the oscillations although when the interface approaches the stable circular configuration the expression an accurate representation of the area of the droplet.



**Figure 13.7.:** Initial interface configuration and computational domain geometry and boundary conditions for the unsteady perturbed droplet case

Grid spacing	area loss (%) colocated	area loss (%) staggered
1/16	17.42	0.8983
1/32	10.02	0.4072
1/64	5.660	0.1962
1/128	3.044	0.1089
1/256	1.608	0.0580

**Table 13.1.:** Percentage area loss of the droplet during the first 4 seconds of the simulation

## 13.4. Flow over a static circular cylinder

The domain chosen for this study matches that of Lai *et al.* [3] for validation purposes, consisting of a cylinder with radius 0.15m centred at  $x = 1.85\text{m}$ ,  $y = 4.0\text{m}$  within a 8m x 8m square domain. The geometry was chosen to be large enough such that the periodic boundary conditions have little impact on the cylinder wake.

Below a critical Reynolds number (of approximately 40) the cylinder wake will be steady

and symmetric. Above this critical Reynolds number vortex shedding will occur resulting in a Von-Kamen vortex sheet trailing behind the cylinder. Vortices are shed from the upper and lower side of the cylinder alternately. The asymmetrical velocity fields result in lift and drag being generated with the drag oscillating at twice the frequency of the lift.

The non-dimensional parameter characterising the vortex shedding is the Strouhal number (Equation 13.2)

$$S = \frac{fL}{V} \quad (13.2)$$

Where  $f$  is the frequency of vortex shedding,  $L$  is the characteristic length of the cylinder and  $V$  is the relative velocity of the cylinder with respect to the flow field.

The cylinder is represented by an immersed interface with each point within the interface anchored to its initial position. The forces at each point are calculated using Equation 13.3, where  $k$  is large ( $\sim 50000$ ). Additional damping ( $\eta$ ) was required.

$$\mathbf{F} = k(\mathbf{X} - \mathbf{X}_0) - \eta \mathbf{U} \quad (13.3)$$

The domain is periodic in both directions. Because of this the flow over the cylinder is imposed in a more indirect method than if pressure and velocity values and gradients are defined at the external boundaries.

In order to impose a uniform flow at the inlet (similarly to Lai *et al.* [3]), the  $u$  and  $v$  velocities of a thin strip (two nodes in width) of nodes at the inlet are set to  $(1.0, 0)$  m/s before each time step. This effectively applies a force source term proportional to the mismatch in speed and results in the inlet  $x$ -velocity remaining at unity in the  $x$  direction during the simulation and stops the wake influencing the inlet  $y$ -velocity.

Simulations were run for Reynolds numbers of 40, 100, 150, 200, and 400. The density of the fluid under consideration was set to unity and the fluid viscosity changed for each simulation to set the desired Reynolds number. The unsteady solutions were run for up to 600 seconds, although in most cases vortex shedding was occurring at a constant rate earlier in the simulation.

A grid independence study was undertaken with transient results generated for a coarse (512x512 nodes) and a medium grid (1024x1024 nodes) for each Reynolds number and an additional high density grid (2048x2048) simulation was run for each Reynolds number to check for convergence.

The calculated Strouhal numbers can be seen in Table 13.2 with a comparison to the results of Lai *et al.* [3] and to experimental results from Roshko *et al.* [5].

The current study's results at both medium and high grid density are in excellent agreement with the experimental data, and with the results of Lai *et al.* [3] who used a similar method.

Grid independence has been further investigated by studying the forces on the cylinder early in the simulation. The calculated results are presented in table Table 13.2.

An experimentally derived relationship between Strouhal number and Reynolds number

Re	Coarse	Medium	Fine	Lai Fine	Experimental
40	-	-	-	-	-
100	0.156	0.163	0.164	0.165	0.164
150	0.1773	0.181	0.181	0.184	0.182
200	0.183	0.189	0.189	0.190	0.190
400	0.214	0.217	0.218	-	-

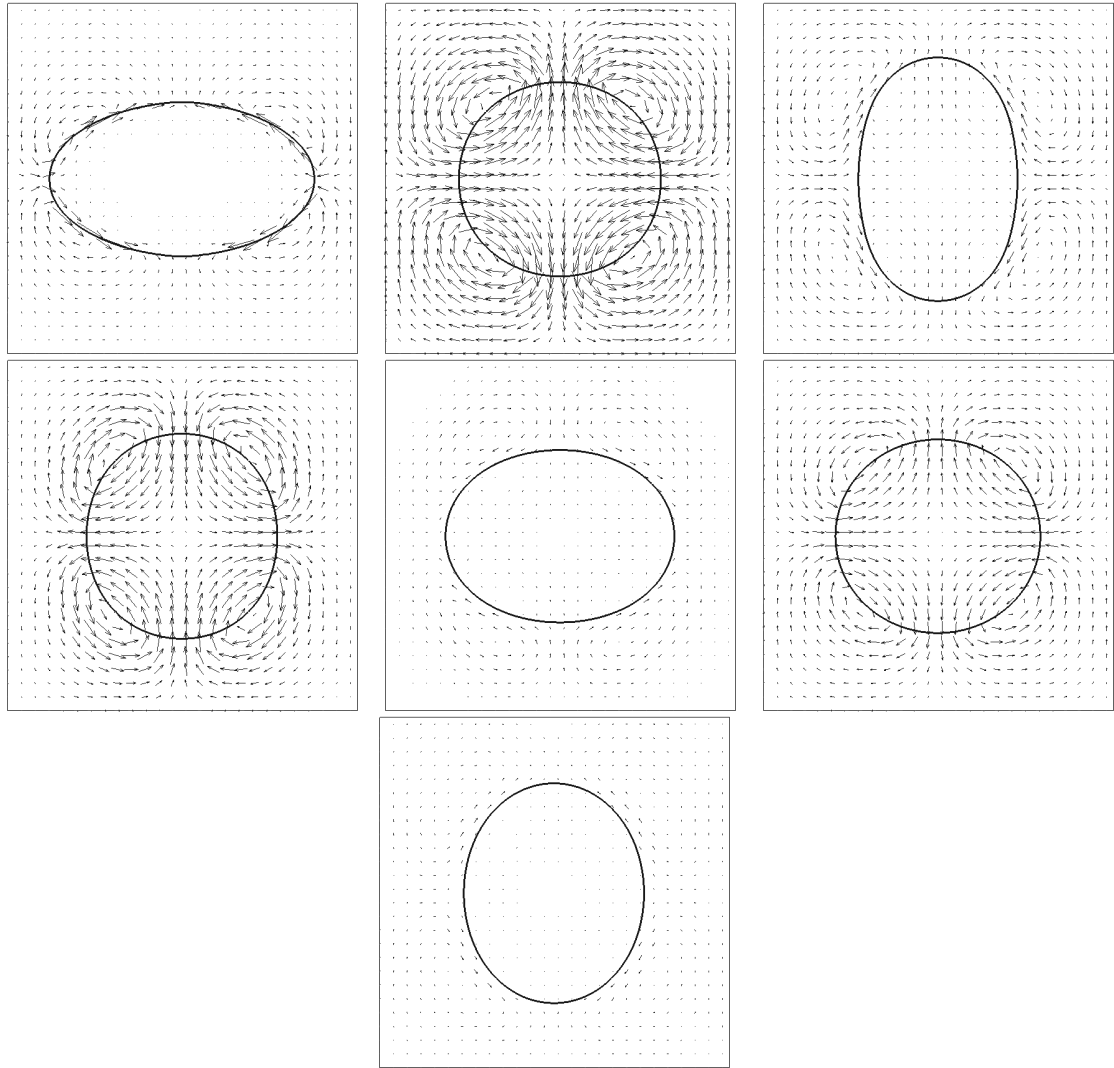
**Table 13.2.:** Strouhal numbers calculated from the current results, Lai *et al.* [3] and the experimental results of Roshko *et al.* [5]

has been found in Williamson *et al.* [4] which is reproduced below

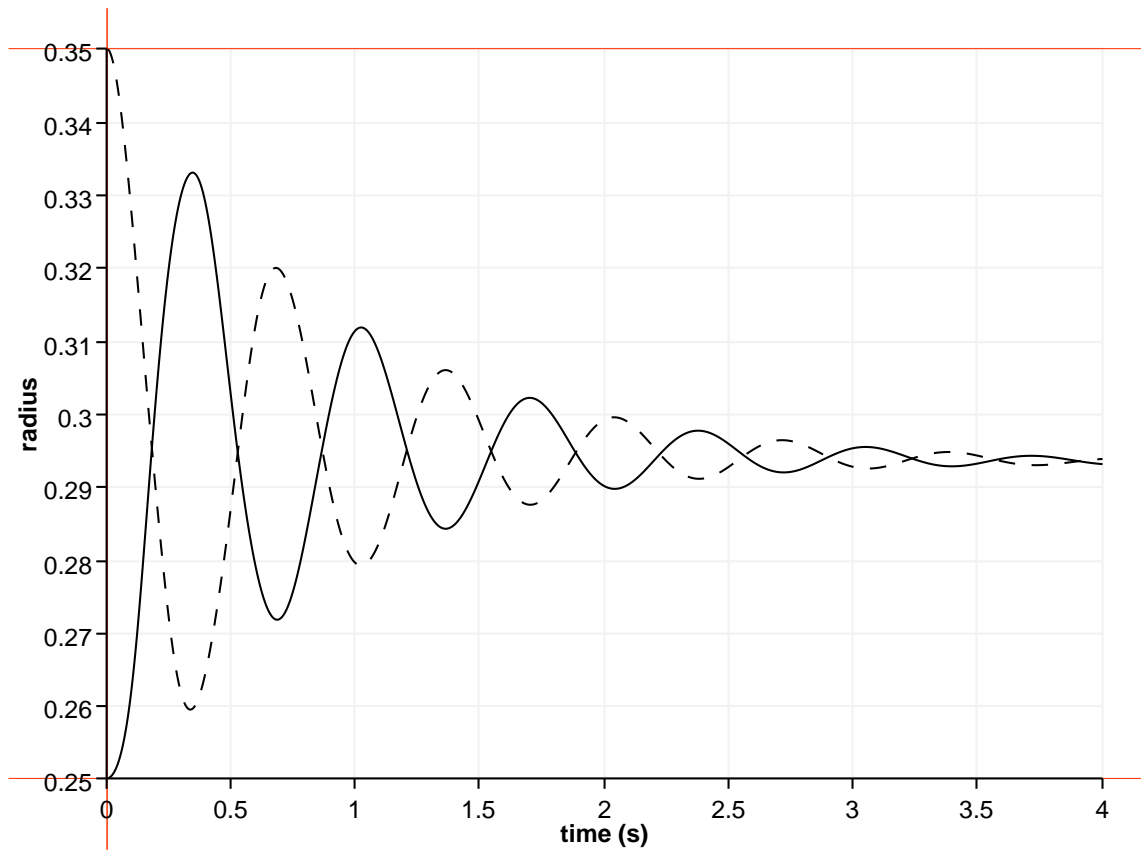
$$S = 0.285 - \frac{1.390}{\sqrt{Re}} + \frac{1.806}{Re}$$

Plotting these results displayed in Table 13.2 shows a good fit with the current data and that of Lai *et al* [3].

Contours of vorticity and the velocity magnitudes are presented in Figure 13.13 Figure 13.14 respectively. As can be seen in the figures for every Reynolds number above 40, the cylinder's wake was found to be asymmetrical and unsteady. For Reynolds numbers above 40 vortex shedding occurred with the spacing decreasing with increasing Reynolds number. The results are presented for the lower resolution mesh, and the cylinder interface is slightly deformable.

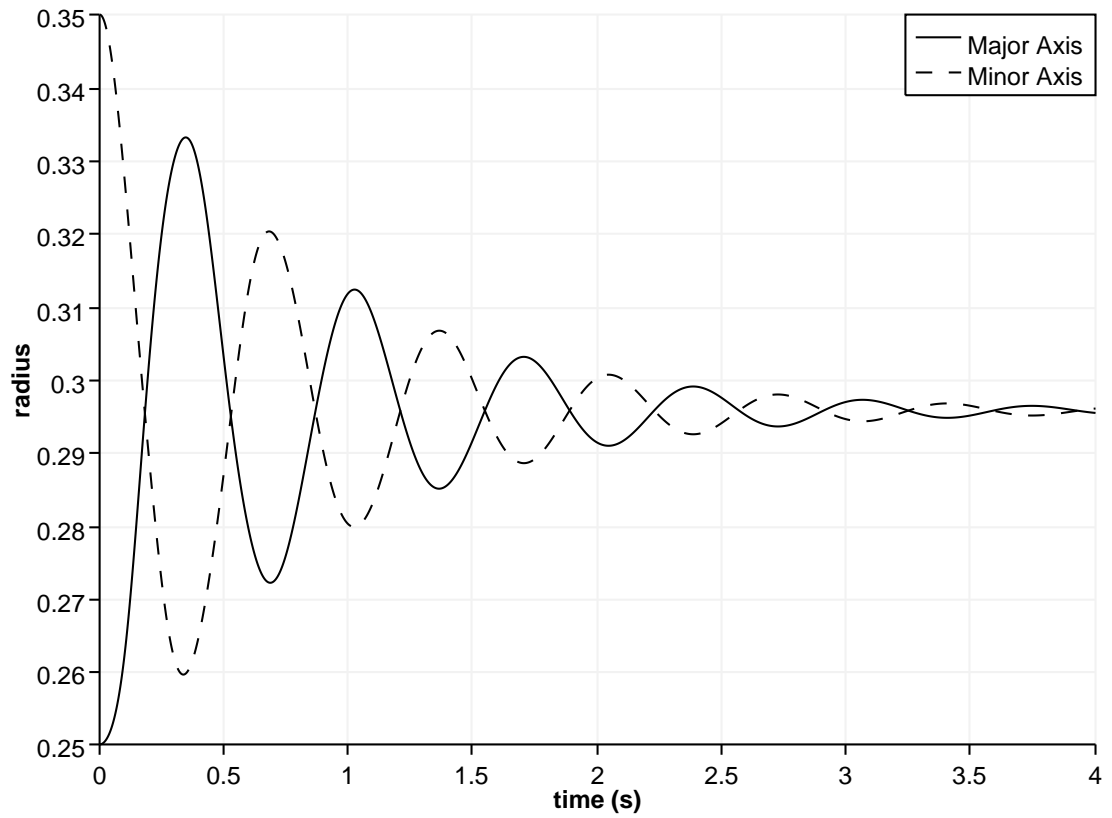


**Figure 13.8.:** Velocity vectors and droplet interfaces showing the dynamic response of the droplet at 0, 0.15, 0.3, 0.45, 0.6, 0.75 and 0.9 seconds.

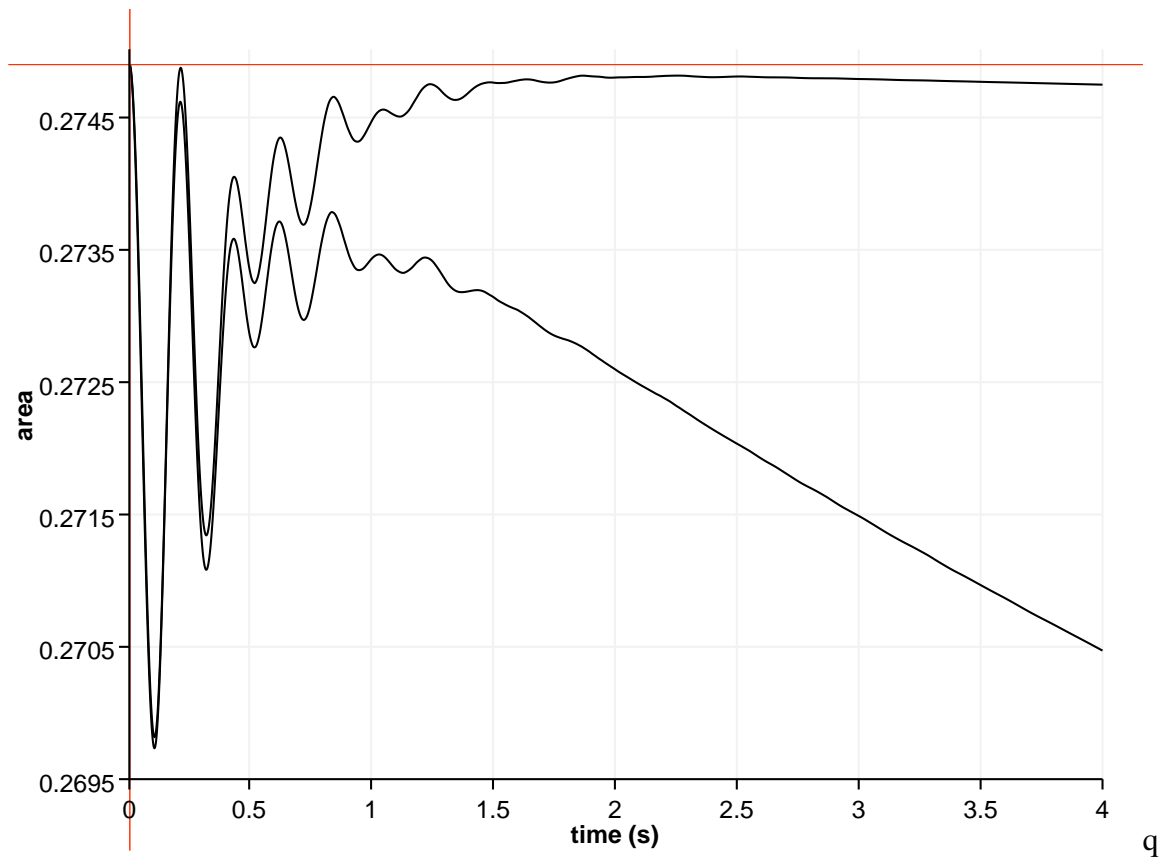


**Figure 13.9.:** Response of the droplet major and minor axes on the 128x128 grid using the colocated finite difference scheme, note the slight downward slope

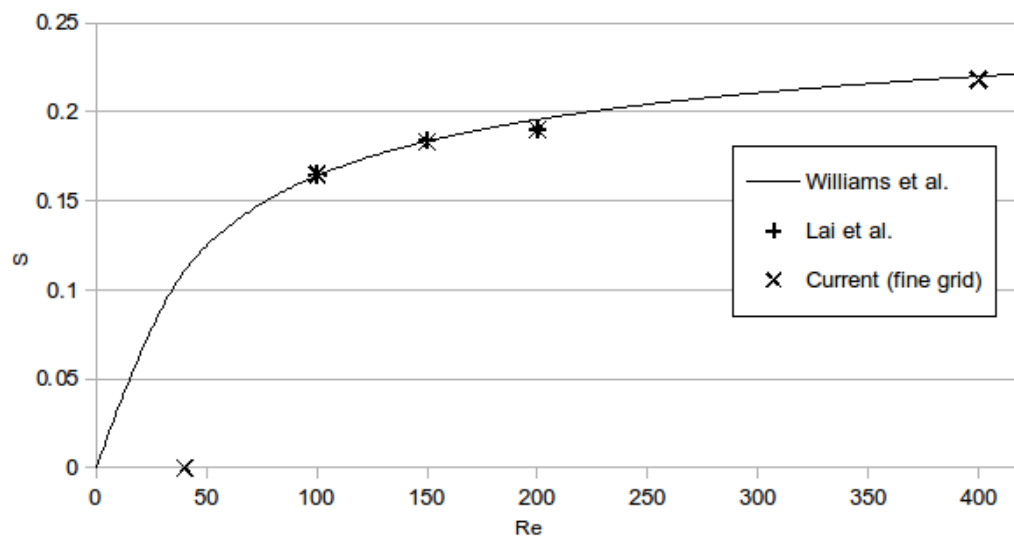




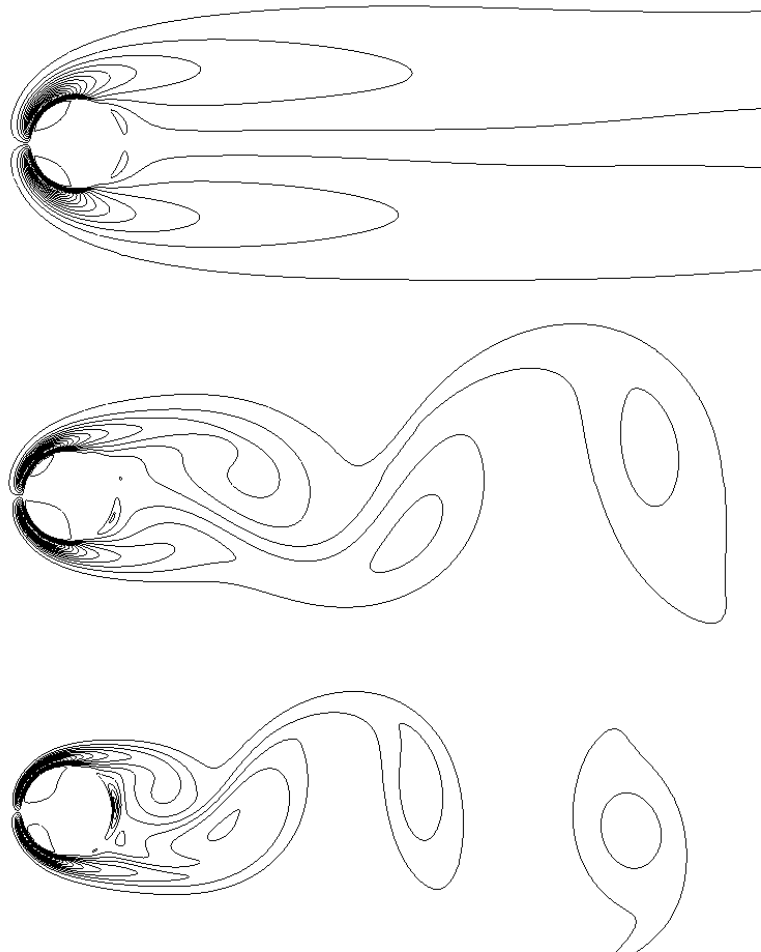
**Figure 13.10.:** Response of the droplet major and minor axes on the 128x128 grid using the staggered (MAC) finite difference scheme



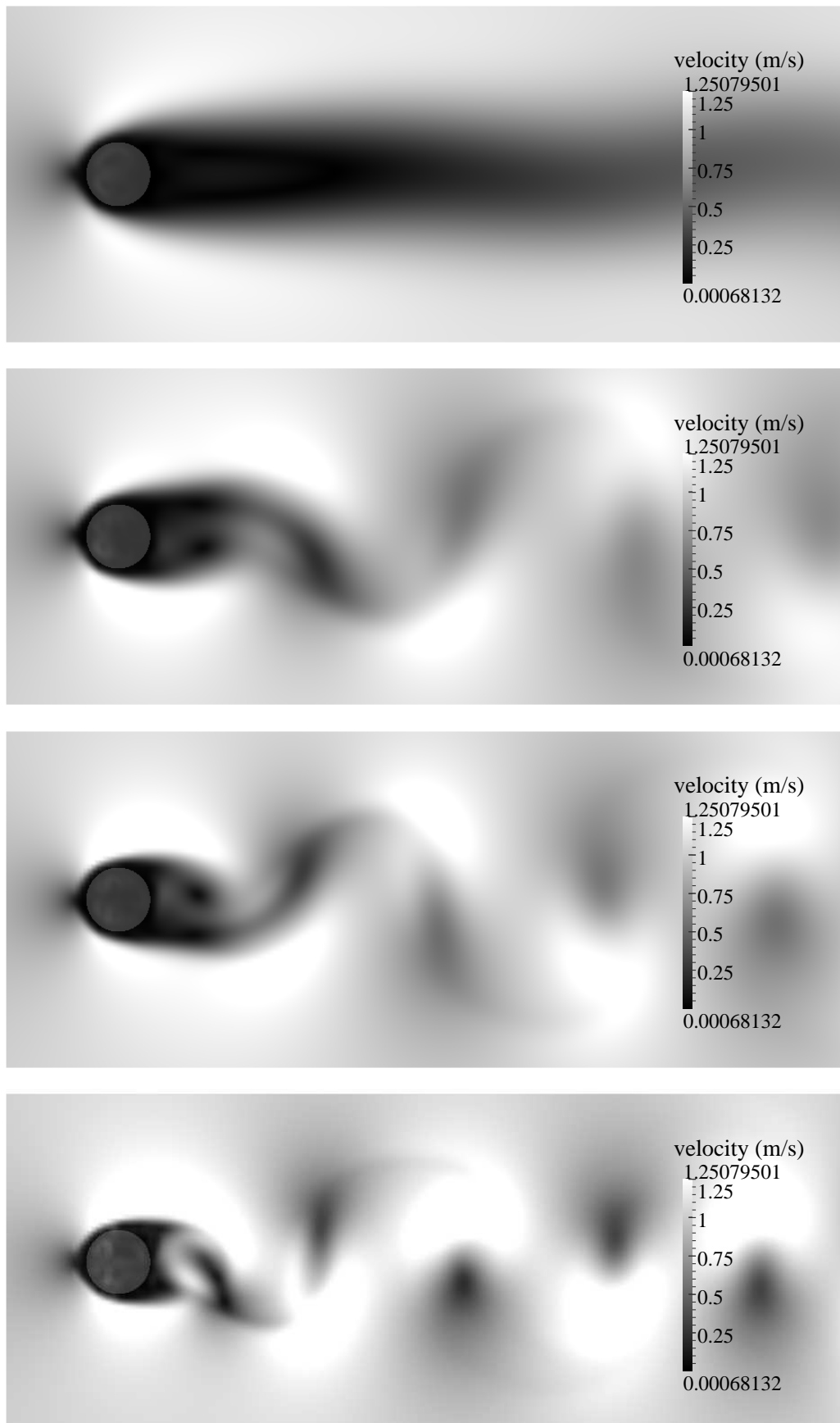
**Figure 13.11.:** Droplet volume assuming the interface to be elliptical during the simulation. The top plot is for the staggered discretisation and shows a significant increase in volume conservation properties compared to the collocated grid (bottom)



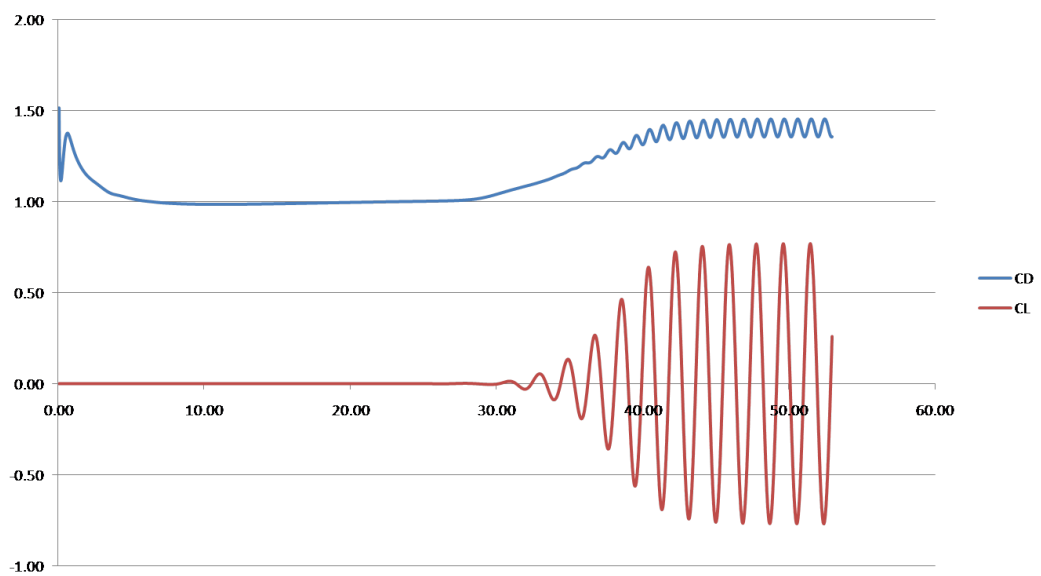
**Figure 13.12.:** Reynolds number vs Strouhal number for the current results, Lai *et al.* [3] and the experimentally derived fitting of Williamson *et al.* [4]



**Figure 13.13.:** Contours of vorticity around a static circular cylinder calculated using the immersed boundary method for Reynolds numbers of 40 (top), 100 & 400 (bottom)



**Figure 13.14.:** Velocity magnitude around a static circular cylinder calculated using the immersed boundary method Reynolds numbers 40,100,150,200 and 400 increasing top to bottom. The immersed cylinder geometry is lightened for clarity



**Figure 13.15.:** Lift and drag coefficient for  $Re = 150$  case

## 14. Limitations and alternative approaches

The immersed boundary method has a number of significant limitations.

Firstly there is a timestep limitation which becomes more severe as more stiffness is introduced to the interface's constitutive equation. The method is only first order accurate local to interface and volume is not conserved adequately either side of an immersed interface. In the case of a colocated discretisation for many typical applications the volume conservation was inadequate to the point where it was difficult to be sure that the method was correctly implemented without cross referencing the results with those generated from third party implementations.

The severe timestep limitations can be overcome in a number of ways.

Firstly the boundary forces can be calculated treating the boundary velocities implicitly as in Taira *et al.* [67] by introducing the regularised delta functions to the pressure equation. In this approach the interface velocities constrained using Lagrangian multiplier in the Poisson equation resulting the pressure equation removing both the divergence in the velocity fields and the error in the velocity fields at the interface.

Another interesting approach that is applicable to coupling both velocities and displacements (i.e. using a direct forcing and continuous forcing approach incorporating structural constitutive models) is that of Cenicerros *et al.* [82, 83] who note that with lagged spreading and interpolating operators, it is possible to construct the Jacobian matrix of the boundary displacement (or velocity) with respect to the boundary force if the fluid domain is periodic and the projection and momentum operators commute.

### 14.1. Implicit solution

The time-step limitation of the explicit scheme makes the solution of problems with stiff interfaces computationally expensive. This can be overcome with the use of an implicit boundary advancement scheme. To make the problem implicit, the boundary advancement equations (Crank-Nicholson or backwards Euler) is rearranged to give

$$G(\mathbf{X}) = \mathbf{X}^{n+1} - \frac{1}{2}\Delta t \mathbf{U}^{n+1} - \left( \mathbf{X}^n + \frac{1}{2}\Delta t \mathbf{U}_b^n \right)$$

$$G(\mathbf{X}) = \mathbf{X}^{n+1} - \Delta t \mathbf{U}^{n+1} - \mathbf{X}^n$$

Ideally, the implicit solution of this problem would start by calculating the Hessian matrix (the second partial derivatives of residuals of the position of each point on the discretised boundary) and use Newton's method to approximate an improvement on the boundary position at the next time step. Due to the influence of every Lagrangian and Eulerian variable at every point for each residual, this would be difficult if not impossible to compute directly. A quasi-Newton method (Nocedal and Wright [78]) in which an approximation to



the Hessian matrix is estimated and updated with each iteration can be used to overcome this.

In the current implementation the BFGS method (Nocedal and Wright [78]) is used, further details can be found in [84]. This method performs the following steps –

1. Initialize the inverse Hessian ( $B_m^n$ ) to the identity matrix.
2. Update the boundary configuration using a form of the Newton equation.

$$\mathbf{X}_{m+1} = \mathbf{X}_m - B_m^n G(\mathbf{X}_m) \quad (14.1)$$

In Equation 14.1  $X$  is a column vector containing the interface positions in both spatial directions, because  $B$  includes the approximations of each and every second partial derivative.

3. Update the approximation of  $B$ . where  $y$  and  $s$  are the change in residual and the boundary position  $\mathbf{X}$  respectively

$$B = B - \frac{Bss^T B}{s^T B s} + \frac{yy^T}{s^T y} \quad (14.2)$$

4. When the residual reduces to the required tolerance, the values of the fields and output of  $G_m(\mathbf{X}_m)$  and  $\mathbf{X}_m$  are accepted as those of the new time-step, and steps 2-3 are repeated until the desired time.

The derivation of Equation 14.2 can be found in Nocedal and Wright [78].

### Strategies for reducing the number of BFGS loops.

It was found that in cases where the boundaries can be considered smooth, the solution is significantly more computationally efficient if the boundary is defined by a (relatively) small number of control points, and the force applied using Lagrangian markers interpolated to a denser mesh from the boundary marker points using cubic splines.

The numerical accuracy and computational workload required depend strongly on the time step selected for use in each time marching scheme. Simulations have been run at a range of different time-steps for both implicit methods.

As the interface position is captured accurately using a relatively small number of control points, running the additional routines required for BFGS minimisation has little impact on the computational cost. The additional cost is due almost entirely to solving the Navier-Stokes equations and spreading, interpolating and re-sampling forces and velocities using delta functions and cubic splines to calculate the residual at each sub-step. This means that the number of sub-steps gives an accurate measure of the increasing computational workload, and can be used to directly compare the implicit and explicit methods.

## 14.2. Solution strategy with the flow solve operator

Ceniceros *et al.* [?, ?] note that the original semi implicit method of Peskin [3] treats the fluid force as follows

$$\mathbf{f} = aHS(\mathbf{X}^{n+1})$$

Where  $S$  is a matrix or a non linear operation relating the interface position to the Lagrangian force at the interface and  $H$  is the spreading operation defined in the previous section. Expressing the fluid solver as  $L_h$  allows the semi implicit method to be expressed as

$$\begin{aligned}\mathbf{u}^{n+1} &= L_h(\mathbf{RHS} + \mathbf{f}) \\ \mathbf{X}^{n+1} &= \mathbf{X}^{n+1} + \Delta t b H^T \mathbf{u}^{n+1} = \frac{\Delta t^2}{\rho} M_n S(\mathbf{X}^{n+1}) + \mathbf{X}^n + \Delta t b H^T L_h \mathbf{RHS}\end{aligned}$$

Where  $M_n$  is the Jacobian matrix relating forces on the interface points to velocities on the interface points. The coefficients of  $M_n$  thus correspond to the velocity obtained by interpolating the velocity values produced at boundary point  $i$  by a spread unit force in the  $x$  or  $y$  direction at boundary point  $j$ . This operator is constructed by evaluating the velocity fields generated by applying a unit force at the origin in the horizontal and vertical direction, spreading these velocities to the Eulerian grid, and then (assuming translation invariance) linearly interpolating the velocity field to the distance between the corresponding points  $i$  and  $j$  from the origin.

Each force (with components in the  $x$  and  $y$  direction) at each node thus results in 4 induced velocities which are stored in  $M_n$  as shown below

$$M = \begin{bmatrix} \left[ \begin{array}{ccc} \frac{\partial U_{x1}}{\partial F_{x1}} & \dots & \frac{\partial U_{x1}}{\partial F_{xn}} \\ \vdots & \ddots & \\ \frac{\partial U_{xn}}{\partial F_{x1}} & & \ddots \\ \frac{\partial U_{y1}}{\partial F_{x1}} & \dots & \frac{\partial U_{y1}}{\partial F_{xn}} \\ \vdots & \ddots & \\ \frac{\partial U_{yn}}{\partial F_{x1}} & & \ddots \end{array} \right] & \left[ \begin{array}{ccc} \frac{\partial U_{x1}}{\partial F_{y1}} & \dots & \frac{\partial U_{x1}}{\partial F_{yn}} \\ \vdots & \ddots & \\ \frac{\partial U_{xn}}{\partial F_{y1}} & & \ddots \\ \frac{\partial U_{y1}}{\partial F_{y1}} & \dots & \frac{\partial U_{y1}}{\partial F_{yn}} \\ \vdots & \ddots & \\ \frac{\partial U_{yn}}{\partial F_{y1}} & & \ddots \end{array} \right] \end{bmatrix}$$

This operator can also be used directly to determine the interface forces required to set the interface velocity, or solve for interface velocities given a force distribution.

## 15. Further work

Creating geometries for immersed boundary method simulations usually consists of discretising analytical functions which usually describes a periodic interface or a fibre resulting in a severely limited and consistent set of typical case studies and validation cases. In order to attempt to make more interesting cases available algorithms were developed and programs written to initialize boundaries from vector (svg) and raster images. In the case of raster images this proved a complex task - generating a smooth equally spaced representative sample of points from arbitrary (but smooth) geometries with minimal user input. It is anticipated that completing this approach would allow the fast generation of cases from medical images.

A number of attempts have been made to alter the discretisation to account for the immersed boundary whilst retaining the convenience of the delta function. The first attempt summed the delta function weighting at points with equal indices and directly altered the equation in these cells. This was applied both as an update loop over the boundary during the implicit solution and by directly altering the equations. This worked in that it assigned the cells to a prescribed value such that when interpolated to the Lagrangian structure the sums of the values at Lagrangian grid points were as prescribed but the values were not smooth over the boundary.

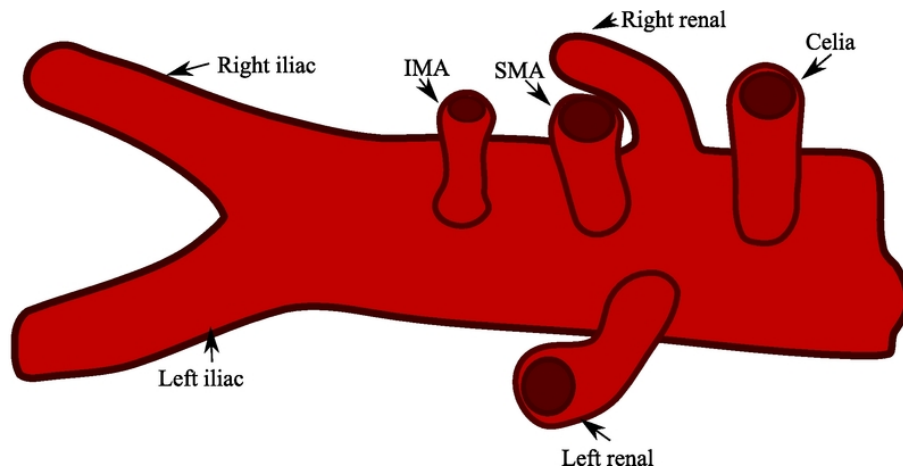
It was decided to use a similar technique solving for all the cells supported by the delta function with the additional constraints that the forcing applied should be consistent with that of forces applied at the Lagrangian points using delta functions. It has been discovered that an underlying problem with approaches of this kind is that the system is overdetermined and numerous approaches to overcoming this shortcoming are currently being investigated.

## **A. Aneurysm simulations using commercial codes**

During the course of this research, various cases have been run designed and discussed in collaboration with surgeons in order to better understand the flow of blood through aneurysms. The studies have progressed from simple two dimensional axisymmetric cases, to transient non-newtonian unsteady flow in complex geometries.

Aortic aneurysms are irreversible, balloon-like dilations in the aorta. This fusiform bulge typically occurring between the renal and iliac arteries, and expand, typically over a period of years, due to gradual wall weakening, and may eventually rupture. Approximately 6% of men and 2% of women develop aortic aneurysms.

The two main treatments options for aortic aneurysms are open surgery, in which a synthetic section is inserted into the aneurysm and sewn in at either end, and endovascular aneurysm repair. Endovascular aneurysm repair consists of the insertion and expansion of a bifurcating (trouser shaped) stent-graft in the aorta, cutting off blood flow to the aneurysm sac and thus preventing rupture or further growth.



**Figure A.1.:** Schematic showing the blood vessels connected to the aorta.

In order to stop stent migration, proximal and distal landing zones of 1.5cm are recommended to transmit the stent's radial force to the aorta. Many patients do not have a sufficiently long neck section, and so are not suitable for this procedure.

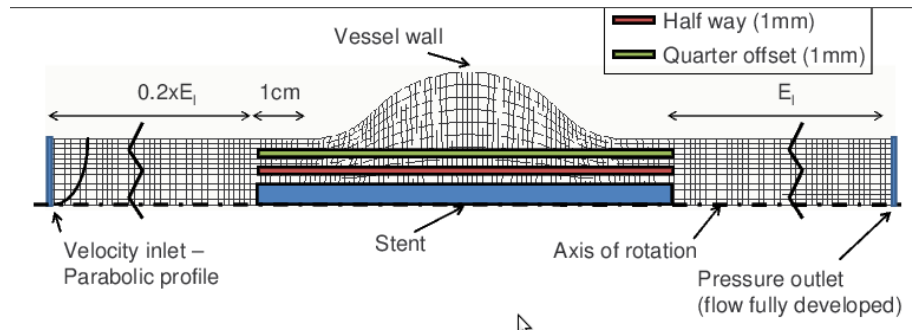
Two dimensional simulations using the the commercial finite volume CFD code FLUENT and the meshing utility GAMBIT have been run to estimate this force, and gain a rough estimate of the flow patterns during deployment to aid the design of stent specialised for these kinds of problems. A brief presentation of some of the results is provided here.

To investigate stent deployment, four different geometries were considered. The aortic aneurysm pre stent insertion, with the undeployed stent, with the half deployed stent and a three-quarter deployed stent in the centre.

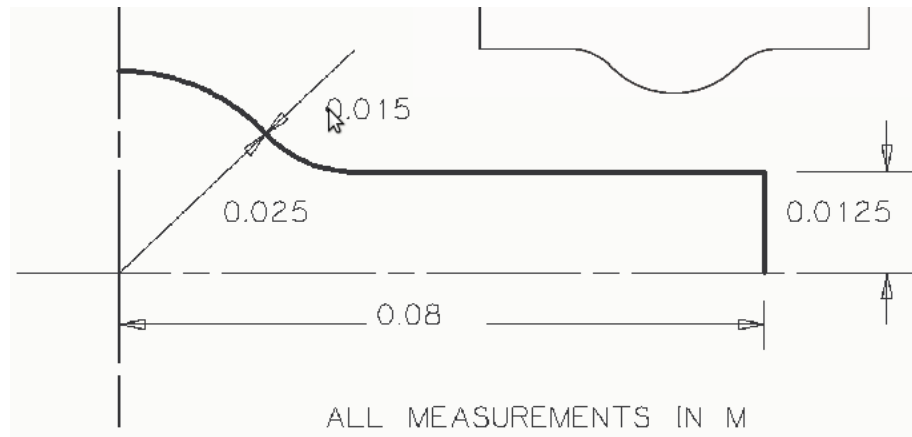
The aortic aneurysm in this study is idealised to an axisymmetric smoothed 1–2–1 expansion–contraction shown in Figure A.3

To calculate the Reynolds number, and so inlet velocity of the study, the flow of blood through the aorta was considered.





**Figure A.2.:** Geometry and boundary conditions for early axisymmetric AAA simulations



**Figure A.3.:** Axisymmetric AAA geometry used in early studies

The heart produces a flow rate of  $8 \times 10^5 \text{ m}^3/\text{s}$  into the aorta [6]. The percentage of flow through each artery is given below in Table 3, along with the calculated percentage of flow rate reaching the inlet of this study, the zone downstream of the renal arteries.

Using the flow rate and the aorta diameter used in this study the time averaged Reynolds number at the neck is 422, and the average velocity  $v_{av}$  is  $0.0554 \text{ m/s}$ . At systole, the Reynolds number increases to approximately three times this time averaged value [6].

Literature gives a range of values for peak and average aortic Reynolds number (depending strongly on the subject and conditions – e.g. there is approximately a 25% increase

Position	Flow rate
Celiac Artery	21%
SMA	21%
Left Renal Artery	15%
Right Renal Artery	15%
Domain Inlet	34%

**Table A.1.:** Typical flow through the aorta and connecting arteries (from [6])

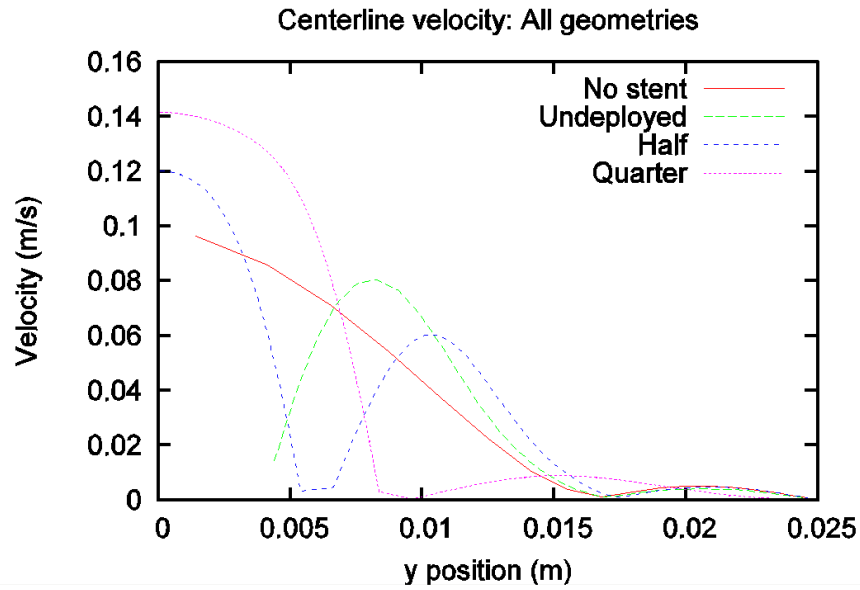
when lying face up as opposed to sitting down). Average values range from 362 to 1053, and at peak conditions from 1950 to 5695 ( from [6, 85, 86, 87] ). For this preliminary study it was decided to take the calculated averaged values to ensure the flow would not be turbulent and to reduce mesh size. The blood viscosity and pressure used in this study are  $4.2e - 3$  and 13.33 kPa and respectively.

A parabolic velocity profile was fitted at the inlet, and downstream pressure outlet boundary conditions were set. The artery/aneurysm wall and the stent (where present) surfaces were set to no-slip walls. The axis about which the domains were axisymmetric was set to an axis boundary condition. To ensure that the flow at outlet was fully developed the downstream section length was set equal to the entrance length  $El$  , which for laminar flow is defined as

$$El = 0.06 * Re * d = 0.06 * 422 * 0.025 = 0.633m$$

The parameters of interest in this study are the wall shear stresses on the aneurysm wall, the force on the stent, and the velocity and pressure fields.

Figure A.6 presents the grid independent mesh with the calculated pressure contours for



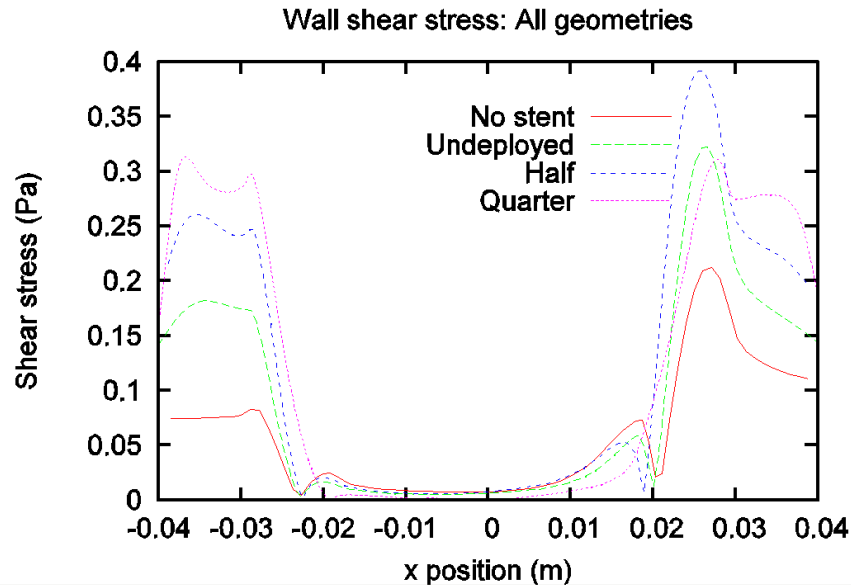
**Figure A.4.:** Centreline velocity plot for 2 dimensional abdominal aortic aneurysm case

the aortic aneurysm geometry with no stent.

The peak pressure values are found on the aneurysm wall at the distal end of the aneurysm, where the main flow reattaches/impinges on the wall. This finding has been observed in literature.

The velocity vectors show a large area of slow recirculating flow within the aneurysm body, this can also be seen in the centreline velocity plots Figure A.4 This recirculation results in the area of low wall shear stress magnitude seen in Figure A.4 around the centre of the aneurysm body ( $x = 0$ ).

The reattachment of the faster flow causes the high wall shear stresses and wall shear stress gradients at the distal end, where the pressure peak was observed. The combination of high wall shear stresses and wall shear stress gradient values (which regulate the vascular endothelial cell's cellular function and so wall material properties) and the peak pressure



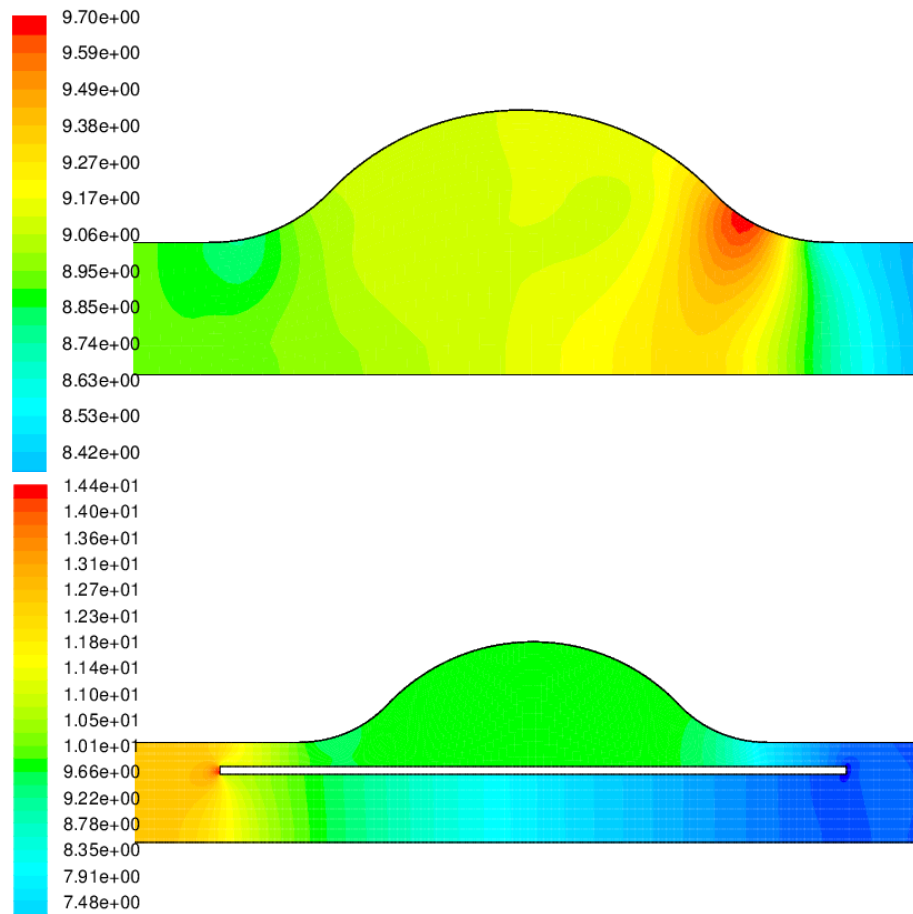
**Figure A.5.:** Wall shear stresses for abdominal aortic aneurysm case

(which results in high structural loads) may result in degeneration of the vascular wall, or further growth in this area.

The inclusion of the stent at a quarter channel width offset from the wall of the healthy aorta section significantly changes the flow field.

The pressure in the aneurysm section is increased by the inclusion of the stent and the sharp localised peak value at the distal end of the aneurysm disappeared. The zone of recirculation is reduced and moved towards the proximal end. However the velocities in this zone are too small for this recirculation to be of much significance.

The centreline velocities show that the velocity, although increased in the radial centre of the aneurysm (as the flow is no longer recirculating) is slower near the aneurysm wall. The peak pressure occurs on the forward face of the stent, but due to its small width, only results in a small pressure force.

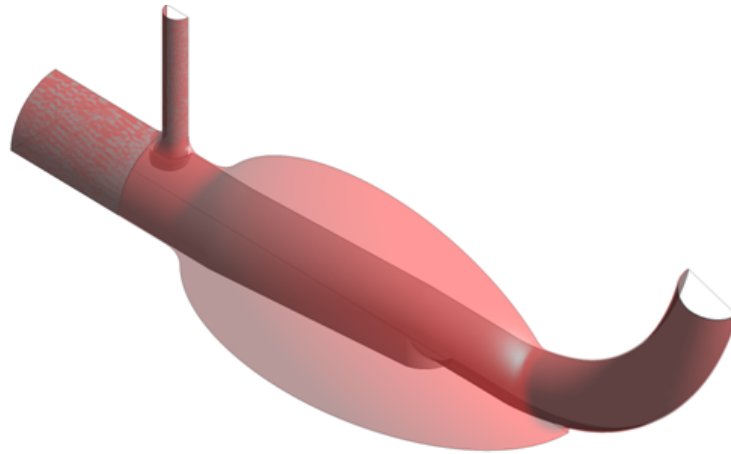


**Figure A.6.:** Pressure contours (kPa) for no stent and quarter deployed stent case

### A.1. 3D Chimney stent

Three dimensional transient simulations have been run for a aortic aneurysm cases incorporating the renal and iliac arteries. Transient simulations have been run to investigate the migration force in this three dimensional model and the flow and structural forces compared.

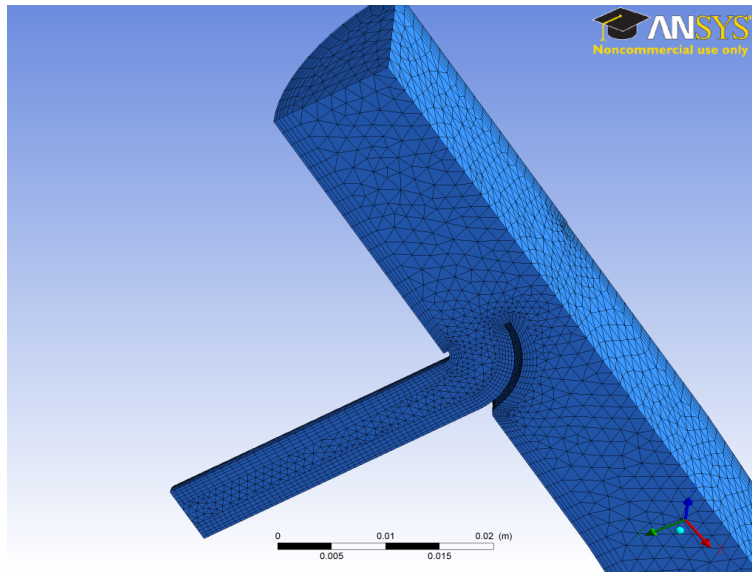
Physically representative quarter geometry such as that shown in Figure A.7 were generated using ANSYS designmodeller for a number of cases.



**Figure A.7.:** 3D chimney stent and aorta geometry used in ANSYS simulations

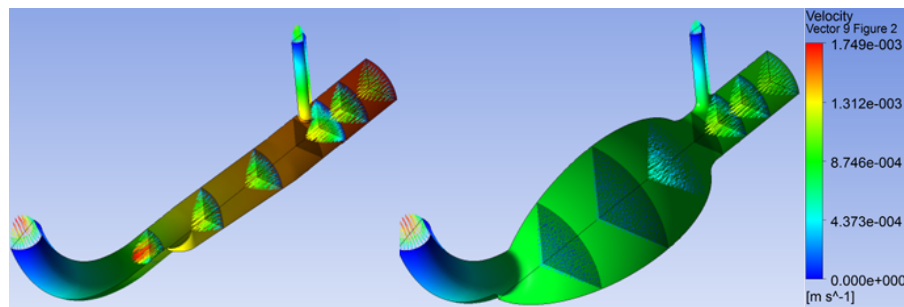
A meshing strategy was chosen to ensure that the boundary layers were adequately modelled and the solutions mesh independent. Figure A.8 shows the coarsest mesh for the stent geometry.

Cycle independent transient results were found after three cardiac cycles. Velocity vectors and pressure contours are plotted below at three interesting stages of the simulation, at systole (Figure A.9), for average flowrate (Figure A.10) and at diastole(Figure A.11), where there is flow reversal. To allow this feature outlet boundary conditions were applied with pressures defined at the iliac and renal outlets. A physiologically representative transient inlet profile was assigned to the inlet using a high order Fourier approximation.

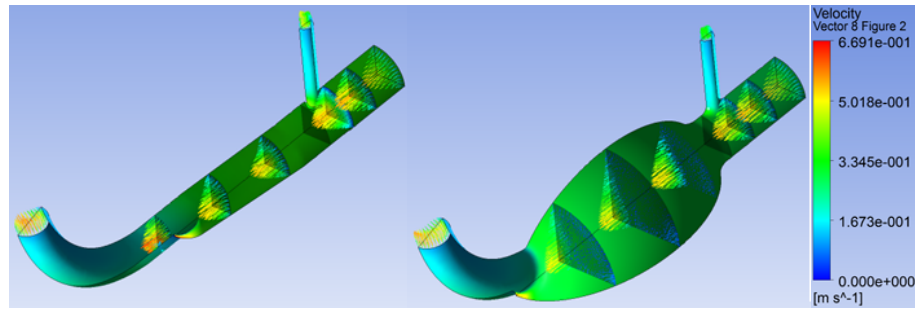


. A fully transient simulation is then compared to steady results to determine the influence of the pulsing flow, determining if the transient nature results in strong oscillations or significantly different flow physics, and finally an investigation into the relative importance of blood rheology is undertaken.

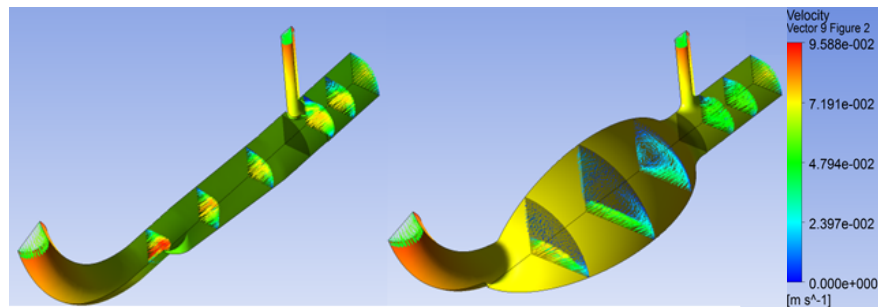
**Figure A.8.:** Coarse mesh for the chimney stent investigation



**Figure A.9.:** Velocity vectors and pressure contours in a chimney stent and untreated aortic aneurysm at highest flow rate



**Figure A.10.:** Velocity vectors and pressure contours in a chimney stent and untreated aortic aneurysm at the average flow rate



**Figure A.11.:** Velocity vectors and pressure contours in a chimney stent and untreated aortic aneurysm during flow reversal



# Bibliography

- [1] Ulrich Schumann and Roland A Sweet. Fast fourier transforms for direct solution of poisson's equation with staggered boundary conditions. *Journal of Computational Physics*, 75(1):123–137, 1988.
- [2] Urmila Ghia, Kirti N Ghia, and CT Shin. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of computational physics*, 48(3):387–411, 1982.
- [3] Ming-Chih Lai and Charles S Peskin. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics*, 160(2):705–719, 2000.
- [4] C.H.K. WILLIAMSON and G.L. BROWN. A series in  $1/\sqrt{Re}$  to represent the strouhal-reynolds number relationship of the cylinder wake. *Journal of Fluids and Structures*, 12(8):1073 – 1085, 1998.
- [5] Anatol Roshko. On the wake and drag of bluff bodies. *Journal of the Aeronautical Sciences (Institute of the Aeronautical Sciences)*, 22(2), 1955.

- [6] JE Moore, STEPHAN E Maier, DAVID N Ku, and PETER Boesiger. Hemodynamics in the abdominal aorta: a comparison of in vitro and in vivo measurements. *Journal of Applied Physiology*, 76(4):1520–1527, 1994.
- [7] Masaaki Shojima, Marie Oshima, Kiyoshi Takagi, Ryo Torii, Motoharu Hayakawa, Kazuhiro Katada, Akio Morita, and Takaaki Kirino. Magnitude and role of wall shear stress on cerebral aneurysm computational fluid dynamic study of 20 middle cerebral artery aneurysms. *Stroke*, 35(11):2500–2505, 2004.
- [8] Masaaki Shojima, Marie Oshima, Kiyoshi Takagi, Ryo Torii, Kazuya Nagata, Ichiro Shirouzu, Akio Morita, and Takaaki Kirino. Role of the bloodstream impacting force and the local pressure elevation in the rupture of cerebral aneurysms. *Stroke*, 36(9):1933–1938, 2005.
- [9] Andrew Molyneux. International subarachnoid aneurysm trial (isat) of neurosurgical clipping versus endovascular coiling in 2143 patients with ruptured intracranial aneurysms: a randomised trial. *The Lancet*, 360(9342):1267–1274, 2002.
- [10] Mustafa Aziz Hatiboglu, Abdusselam Altunkaynak, Mehmet Ozger, Ahmet Celal Iplikcioglu, Murat Cosar, and Namigar Turgut. A predictive tool by fuzzy logic for outcome of patients with intracranial aneurysm. *Expert Systems with Applications*, 37(2):1043–1049, 2010.
- [11] Martin Sandve Alnæs, Jørgen Isaksen, Kent-André Mardal, Bertil Romner, Michael K Morgan, and Tor Ingebrigtsen. Computation of hemodynamics in the circle of willis. *Stroke*, 38(9):2500–2505, 2007.
- [12] György Paál, Adam Ugron, István Szikora, and Imre Bojtár. Flow in simplified and

- real models of intracranial aneurysms. *International journal of heat and fluid flow*, 28(4):653–664, 2007.
- [13] Loic Bousset, Vitaliy Rayz, Charles McCulloch, Alastair Martin, Gabriel Acevedo-Bolton, Michael Lawton, Randall Higashida, Wade S Smith, William L Young, and David Saloner. Aneurysm growth occurs at region of low wall shear stress patient-specific correlation of hemodynamics and growth in a longitudinal study. *Stroke*, 39(11):2997–3002, 2008.
- [14] Liang-Der Jou, Christopher M Quick, William L Young, Michael T Lawton, Randall Higashida, Alastair Martin, and David Saloner. Computational approach to quantifying hemodynamic forces in giant cerebral aneurysms. *American Journal of Neuroradiology*, 24(9):1804–1810, 2003.
- [15] I Chatziprodromou, A Tricoli, D Poulikakos, and Y Ventikos. Haemodynamics and wall remodelling of a growing cerebral aneurysm: a computational model. *Journal of biomechanics*, 40(2):412–426, 2007.
- [16] I Chatziprodromou, D Poulikakos, and Y Ventikos. On the influence of variation in haemodynamic conditions on the generation and growth of cerebral aneurysms and atherogenesis: A computational model. *Journal of biomechanics*, 40(16):3626–3640, 2007.
- [17] C Alberto Figueroa, Seungik Baek, Charles A Taylor, and Jay D Humphrey. A computational framework for fluid–solid-growth modeling in cardiovascular simulations. *Computer methods in applied mechanics and engineering*, 198(45):3583–3602, 2009.

- [18] Alvaro Valencia and Francisco Solis. Blood flow dynamics and arterial wall interaction in a saccular aneurysm model of the basilar artery. *Computers & structures*, 84(21):1326–1337, 2006.
- [19] Alvaro Valencia, Hernan Morales, Rodrigo Rivera, Eduardo Bravo, and Marcelo Galvez. Blood flow dynamics in patient-specific cerebral aneurysm models: the relationship between wall shear stress and aneurysm area index. *Medical engineering & physics*, 30(3):329–340, 2008.
- [20] Ryo Torii, Marie Oshima, Toshio Kobayashi, Kiyoshi Takagi, and Tayfun E Tezduyar. Fluid–structure interaction modeling of blood flow and cerebral aneurysm: significance of artery and aneurysm shapes. *Computer Methods in Applied Mechanics and Engineering*, 198(45):3613–3621, 2009.
- [21] Ryo Torii, Marie Oshima, Toshio Kobayashi, Kiyoshi Takagi, and Tayfun E Tezduyar. Influence of wall elasticity in patient-specific hemodynamic simulations. *Computers & Fluids*, 36(1):160–168, 2007.
- [22] Juan C Lasheras. The biomechanics of arterial aneurysms. *Annu. Rev. Fluid Mech.*, 39:293–319, 2007.
- [23] JD Humphrey and PB Canham. Structure, mechanical properties, and mechanics of intracranial saccular aneurysms. *Journal of elasticity and the physical science of solids*, 61(1-3):49–81, 2000.
- [24] Satoshi Tateshima, Kazuo Tanishita, Hiroyoshi Omura, James Sayre, J Pablo Villablanca, Neil Martin, and Fernando Vinuela. Intra-aneurysmal hemodynamics in a large middle cerebral artery aneurysm with wall atherosclerosis. *Surgical neurology*, 70(5):454–462, 2008.

- [25] Nicolaas Westerhof, Nikos Stergiopulos, and Mark IM Noble. *Snapshots of hemodynamics: an aid for clinical research and graduate education*. Springer, 2010.
- [26] Oguz K Baskurt and Herbert J Meiselman. Blood rheology and hemodynamics. In *Seminars in thrombosis and hemostasis*, volume 29, pages 435–450. New York: Stratton Intercontinental Medical Book Corporation, c1974-, 2003.
- [27] JW Prothero and AC Burton. The physics of blood flow in capillaries: Ii. the capillary resistance to flow. *Biophysical Journal*, 2(2):199–212, 1962.
- [28] HS Lew and YC Fung. Plug effect of erythrocytes in capillary blood vessels. *Biophysical Journal*, 10(1):80–99, 1970.
- [29] YI Cho and KR Kensey. Effects of the non-newtonian viscosity of blood on flows in a diseased arterial vessel. part 1: Steady flows. *Biorheology*, 28(3-4):241–262, 1990.
- [30] H Tanaka, N Fujita, T Enoki, K Matsumoto, Y Watanabe, K Murase, and H Nakamura. Relationship between variations in the circle of willis and flow rates in internal carotid and basilar arteries determined by means of magnetic resonance imaging with semiautomated lumen segmentation: reference data from 125 healthy volunteers. *American journal of neuroradiology*, 27(8):1770–1775, 2006.
- [31] Robert Fox, Alan McDonald, and Philip Pritchard. Fluid mechanics. 8th. 1, 2011.
- [32] WE Stehbens. Turbulence of blood flow. *Experimental Physiology*, 44(1):110–117, 1959.
- [33] Charles S Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3):220–252, 1977.

- [34] Rajat Mittal and Gianluca Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261, 2005.
- [35] Anna-Karin Tornberg and Björn Engquist. Numerical approximations of singular source terms in differential equations. *Journal of Computational Physics*, 200(2):462–488, 2004.
- [36] Kayne M Arthurs, Leon C Moore, Charles S Peskin, E Bruce Pitman, and HE Layton. Modeling arteriolar flow and mass transport using the immersed boundary method. *Journal of Computational Physics*, 147(2):402–440, 1998.
- [37] John M Stockie and Brian R Wetton. Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes. *Journal of Computational Physics*, 154(1):41–64, 1999.
- [38] Charles S Peskin. The immersed boundary method. *Acta numerica*, 11(0):479–517, 2002.
- [39] Alexandre M Roma, Charles S Peskin, and Marsha J Berger. An adaptive version of the immersed boundary method. *Journal of computational physics*, 153(2):509–534, 1999.
- [40] Rainald Löhner, Juan R Cebal, Fernando E Camelli, S Appanaboyina, Joseph D Baum, Eric L Mestreau, and Orlando A Soto. Adaptive embedded and immersed unstructured grid techniques. *Computer Methods in Applied Mechanics and Engineering*, 197(25):2173–2197, 2008.
- [41] Yongsam Kim and Charles S Peskin. Penalty immersed boundary method for an elastic boundary with mass. *Physics of Fluids*, 19:053103, 2007.

- [42] Duc-Vinh Le, J White, J Peraire, KM Lim, and BC Khoo. An implicit immersed boundary method for three dimensional fluid membrane interactions. *Journal of Computational Physics*, 228(22):8427–8445, 2009.
- [43] DV Le, BC Khoo, and KM Lim. An implicit-forcing immersed boundary method for simulating viscous flows in irregular domains. *Computer methods in applied mechanics and engineering*, 197(25):2119–2130, 2008.
- [44] A Bhalla, Boyce E Griffith, Neelesh A Patankar, and Aleksandar Donev. An immersed boundary method for reaction-diffusion problems. *arXiv preprint arXiv:1306.3159*, 2013.
- [45] Amneet Pal Singh Bhalla, Boyce E Griffith, Neelesh A Patankar, and Aleksandar Donev. A minimally-resolved immersed boundary model for reaction-diffusion problems. *The Journal of chemical physics*, 139(21):214112, 2013.
- [46] Amneet Pal Singh Bhalla, Rahul Bale, Boyce E Griffith, and Neelesh A Patankar. A unified mathematical framework and an adaptive numerical method for fluid-structure interaction with rigid, deforming, and elastic bodies. *J. Comput. Physics*, 250:446–476, 2013.
- [47] Ming-Jyh Chern, Wei-Cheng Shiu, and Tzyy-Leng Horng. Immersed boundary modeling for interaction of oscillatory flow with cylinder array under effects of flow direction and cylinder arrangement. *Journal of Fluids and Structures*, 43:325–346, 2013.
- [48] Wei-Fan Hu, Yongsam Kim, and Ming-Chih Lai. An immersed boundary method for simulating the dynamics of three-dimensional axisymmetric vesicles in navier–stokes flows. *Journal of Computational Physics*, 257:670–686, 2014.

- [49] Shizhao Wang, Guowei He, and Xing Zhang. Parallel computing strategy for a flow solver based on immersed boundary method and discrete stream-function formulation. *Computers & Fluids*, 88:210–224, 2013.
- [50] Iman Borazjani, Liang Ge, Trung Le, and Fotis Sotiropoulos. A parallel overset-curvilinear-immersed boundary framework for simulating complex 3d incompressible flows. *Computers & fluids*, 77:76–96, 2013.
- [51] Antonio J Gil, A Arranz Carreño, Javier Bonet, and O Hassan. An enhanced immersed structural potential method for fluid–structure interaction. *Journal of Computational Physics*, 250:178–205, 2013.
- [52] D Goldstein, R Handler, and L Sirovich. Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics*, 105(2):354–366, 1993.
- [53] Eric Brown-Dymkoski, Nurlybek Kasimov, and Oleg V Vasilyev. A characteristic based volume penalization method for general evolution problems applied to compressible viscous flows. *Journal of Computational Physics*, 262:344–357, 2014.
- [54] Remi Abgrall, Héloïse Beaugendre, and Cécile Dobrzynski. An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques. *Journal of Computational Physics*, 257:83–101, 2014.
- [55] J Mohd-Yusof. Development of immersed boundary methods for complex geometries. *Center for Turbulence Research Annual Research Briefs*, 105:325–336, 1998.
- [56] Markus Uhlmann. An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*, 209(2):448–476, 2005.



- [57] Alfredo Pinelli, IZ Naqavi, Ugo Piomelli, and Julien Favier. Immersed-boundary methods for general finite-difference and finite-volume navier–stokes solvers. *Journal of Computational Physics*, 229(24):9073–9091, 2010.
- [58] Kameswararao Anupindi, Yann Delorme, Dinesh A Shetty, and Steven H Frankel. A novel multiblock immersed boundary method for large eddy simulation of complex arterial hemodynamics. *Journal of computational physics*, 254:200–218, 2013.
- [59] Yu-Heng Tseng and Joel H Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of computational physics*, 192(2):593–623, 2003.
- [60] Jinmo Lee and Donghyun You. An implicit ghost-cell immersed boundary method for simulations of moving body problems with control of spurious force oscillations. *Journal of Computational Physics*, 233:295–314, 2013.
- [61] Junjie Xia, Kun Luo, and Jianren Fan. A ghost-cell based high-order immersed boundary method for inter-phase heat transfer simulation. *International Journal of Heat and Mass Transfer*, 75:302–312, 2014.
- [62] Matthias Meinke, Lennart Schneiders, Claudia Günther, and Wolfgang Schröder. A cut-cell method for sharp moving boundaries in cartesian grids. *Computers & Fluids*, 85:135–142, 2013.
- [63] Jón Tómas Grétarsson and Ron Fedkiw. Fully conservative leak-proof treatment of thin solid structures immersed in compressible fluids. *Journal of Computational Physics*, 245:160–204, 2013.

- [64] Randall J Leveque and Zhilin Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31(4):1019–1044, 1994.
- [65] Christoph Brehm, Christoph Hader, and HF Fasel. Novel immersed boundary/interface method for the compressible navier-stokes equations. *AIAA*, 1110:2012, 2012.
- [66] Zhilin Li and Kazufumi Ito. *The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains*, volume 33. Siam, 2006.
- [67] Kunihiko Taira and Tim Colonius. The immersed boundary method: a projection approach. *Journal of Computational Physics*, 225(2):2118–2137, 2007.
- [68] Barbara Bigot, Thomas Bonometti, Laurent Lacaze, and Olivier Thual. A simple immersed-boundary method for solid–fluid interaction in constant-and stratified-density flows. *Computers & Fluids*, 97:126–142, 2014.
- [69] H Uddin, RMJ Kramer, and Carlos Pantano. A cartesian-based embedded geometry technique with adaptive high-order finite differences for compressible flow around complex geometries. *Journal of Computational Physics*, 262:379–407, 2014.
- [70] Zhi-Gang Feng and Efstathios E Michaelides. The immersed boundary-lattice boltzmann method for solving fluid–particles interaction problems. *Journal of Computational Physics*, 195(2):602–628, 2004.
- [71] J Wu and Chang Shu. Implicit velocity correction-based immersed boundary-lattice boltzmann method and its applications. *Journal of Computational Physics*, 228(6):1963–1979, 2009.

- [72] Kosuke Suzuki and Takaji Inamuro. A higher-order immersed boundary-lattice boltzmann method using a smooth velocity field near boundaries. *Computers & Fluids*, 76:105–115, 2013.
- [73] Julien Favier, Alistair Revell, and Alfredo Pinelli. A lattice boltzmann–immersed boundary method to simulate the fluid interaction with moving and slender flexible objects. *Journal of Computational Physics*, 261:145–161, 2014.
- [74] PN Watton, XY Luo, X Wang, GM Bernacca, P Molloy, and DJ Wheatley. Dynamic modelling of prosthetic chorded mitral valves using the immersed boundary method. *Journal of biomechanics*, 40(3):613–626, 2007.
- [75] John Crank and Phyllis Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 43, pages 50–67. Cambridge Univ Press, 1947.
- [76] Joel H Ferziger and Milovan Perić. *Computational methods for fluid dynamics*, volume 3. Springer Berlin, 2002.
- [77] J Blair Perot. An analysis of the fractional step method. *Journal of Computational Physics*, 108(1):51–58, 1993.
- [78] Jorge Nocedal and Stephen J Wright. *Numerical optimization*, volume 2. Springer New York, 1999.
- [79] Alexandre Joel Chorin. Numerical solution of the navier-stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.

- [80] John Kim and Parviz Moin. Application of a fractional-step method to incompressible navier-stokes equations. *Journal of computational physics*, 59(2):308–323, 1985.
- [81] John M Stockie and Brian TR Wetton. Stability analysis for the immersed fiber problem. *SIAM Journal on Applied Mathematics*, 55(6):1577–1591, 1995.
- [82] Hector D Cenicerros and Jordan E Fisher. A fast, robust, and non-stiff immersed boundary method. *Journal of computational physics*, 230(12):5133–5153, 2011.
- [83] Hector D Cenicerros, Jordan E Fisher, and Alexandre M Roma. Efficient solutions to robust, semi-implicit discretizations of the immersed boundary method. *Journal of Computational Physics*, 228(19):7137–7158, 2009.
- [84] Brian P Flannery, William H Press, Saul A Teukolsky, and William Vetterling. Numerical recipes in c. *Press Syndicate of the University of Cambridge, New York*, 1992.
- [85] CK Chong and TV How. Flow patterns in an endovascular stent-graft for abdominal aortic aneurysm repair. *Journal of biomechanics*, 37(1):89–97, 2004.
- [86] CJ Egelhoff, RS Budwig, DF Elger, TA Khraishi, and KH Johansen. Model studies of the flow in abdominal aortic aneurysms during resting and exercise conditions. *Journal of biomechanics*, 32(12):1319–1329, 1999.
- [87] Denz Lee and JY Chen. Numerical simulation of steady flow fields in a model of abdominal aorta with its peripheral branches. *Journal of biomechanics*, 35(8):1115–1122, 2002.