DEVELOPMENT OF FLUID-SOLID INTERACTION (FSI)

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE & ENGINEERING

2017

By
Jesus Ernesto De La Peña-Cortes
School of Mechanical, Aerospace and Civil Engineering

Contents

Final word count:$\approx$ 45000

List of Algorithms

List of Figures

List of Tables

Nomenclature

Acronyms

ALE Arbitrary Lagrangian-Eulerian

CM Control mass

CFD Computational Fluid Dynamics

CGNS CFD General Notation System

COG Rigid-body Centre of gravity

CV Control volume

FSI Fluid Solid Interaction

FVM Finite Volume Method

MFBI Mass Flux Based Interpolation

QUICK Quadratic Upwind Interpolation for Convective Kinematics

RANS Reynlods-Averaged Navier-Stokes Equations

SCL Space Conservation Law

SI Standard Interpolation

SIMPLE Semi-Implicit Method for Pressure Linked Equations

VIV Vortex Induced Vibrations

Greek Symbols

$\boldsymbol{\alpha}$	Rigid-body angular acceleration vector	$\frac{rad}{s^2}$
α	Relaxation factor	–
γ	Intermediate factor used in adaptive relaxation	–
$\delta\Omega_c$	Volume swept by any control volume face during a time interval	m^3

μ	Fluid dynamic viscosity	$\frac{kg}{m \cdot s}$
ρ	Fluid density	$\frac{kg}{m^3}$
ϕ	Conserved scalar quantity	–
$\boldsymbol{\omega}$	Rigid-body angular velocity vector	$\frac{rad}{s}$
Ω	Computational cell volume	m^3

Latin Symbols

Δt	Time step	s
$\mathbf{a}$	Rigid-body centre of gravity acceleration vector	$\frac{m}{s^2}$
A	Coefficients for the discretised scalar transport equations	$\frac{kg}{s}$
$\mathbf{F}$	Resultant force vector acting on the rigid-body	N
$\mathbf{I}$	Rigid-body inertia tensor, unit tensor	$kg \cdot m^2$, –
$\dot{m}$	Mass flow rate through a cell face	$\frac{kg}{s}$
m	Rigid-body mass	kg
$\mathbf{n}$	Unit normal vector	–
N	Interpolation factors	–
P	Pressure field	$\frac{N}{m^2}$
p	Observed order of accuracy	–
q	Quaternion representing rotational motion	–
$Q_{\Omega_{CV}}$	Volumetric sources	$\frac{N}{m^3}$
Q_p	Resulting known forcing term of the discretised set of equations	N
$\mathbf{Q_s}$	Surface sources	$\frac{N}{m^2}$
$\mathbf{r}$	Position vector from the origin of the inertial coordinate system to any face centre	m
R	Rotation Matrix	–
R_e	Reynolds number	–
$\mathbf{r}_f$	Position vector from the centre of gravity of the rigid-body to the boundary face centre	m

S	Cell face area	m^2
$\mathbf{T}$	Resultant torque vector acting on the rigid-body, molecular stress tensor $N \cdot m$,	$\frac{N}{m^2}$
t	Time	s
$\mathbf{U}$	Fluid velocity vector	$\frac{m}{s}$
$\mathbf{U_g}$	Body wall velocity vector	$\frac{m}{s}$
$\mathbf{V}$	Rigid-body centre of gravity velocity vector	$\frac{m}{s}$
$\mathbf{x}$	Spatial coordinate vector	m
y	Placeholder representing any variable for the kinetimatic state of a rigid-body	$-$

Subscripts & Superscripts

b	Rigid-body attached reference frame
bf	Boundary face
c, f	Any face of a given control volume
$corr$	Corrector
I	Global intertial reference frame
l	Any adjacent neighbouring cell to the current control volume
n	Current time step, scalar value, or iteration
o	Initial or previous time value
oo	Previous previous time value
$pred$	Predictor
T	Transpose

Abstract

DEVELOPMENT OF FLUID-SOLID INTERACTION (FSI)
Jesus Ernesto De La Peña-Cortes
The University of Manchester, Doctor of Philosophy, December 2017

This work extends a previously developed finite-volume overset-grid fluid flow solver to enable the characterisation of rigid-body-fluid interaction problems.

To this end, several essential components have been developed and blended together. The inherent time-dependent nature of fluid-solid interaction problems is captured through the laminar transient incompressible Navier-Stokes equations for the fluid, and the Euler-Newton equations for rigid-body motion. First and second order accurate time discretisation schemes have been implemented for the former, whereas second and third order accurate time discretisation schemes have been made available for the latter.

Without doubt the main advantage the overset-grid method offers regarding moving entities is the avoidance of the time consuming grid regeneration step, and the resulting grid distortion that can often cause numerical stability problems in the solution of the flow equations. Instead, body movement is achieved by the relative motion of a body fitted grid over a suitable background mesh. In this case, the governing equations of fluid flow are formulated using a Lagrangian, Eulerian, or hybrid flow description via the Arbitrary Lagrangian-Eulerian method. This entails the need to guarantee that mesh motion shall not disturb the flow field. With this in mind, the space conservation law has been hard-coded. The compliance of the space conservation law has the added benefit of preventing spurious mass sources from appearing due to mesh deformation.

In this work, two-way fluid-solid interaction problems are solved via a partitioned approach. Coupling is achieved by implementing a Picard iteration algorithm. This allows for flexible degree of coupling specification by the user. Furthermore, if strong coupling is desired, three variants of interface under-relaxation can be chosen to mitigate stability issues and to accelerate convergence. These include fixed, or two variants of Aitken's adaptive under-relaxation factors. The software also allows to solve for one-way fluid-solid interaction problems in which the motion of the solid is prescribed.

Verification of the core individual components of the software is carried out through the powerful method of manufactured solutions (MMS). This purely mathematically based exercise provides a picture of the order of accuracy of the implementation , and serves as a filter for coding errors which can be virtually impossible to detect by other means.

Three instances of one-way fluid-solid interaction cases are compared with simulation results either from the literature, or from the OpenFOAM® package. These include: flow within a piston cylinder assembly, flow induced by two oscillating cylinders, and flow induced by two rectangular plates exhibiting general planar motion.

Three cases pertaining to the class of two-way fluid-interaction problems are presented. The flow generated by the free fall of a cylinder under the action of gravity is computed with the aid of an intermediate 'motion tracking' grid. The solution is compared with the one obtained using a vorticity based particle solver for validation purposes. Transverse vortex induced vibrations (VIV) of a circular cylinder immersed in a fluid, and subject to a stream are compared with experimental data. Finally, the fluttering motion of a rectangular plate under different scenarios is analysed.

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 other institute of learning.

Copyright

Acknowledgements

I would like to thank Prof. Hector Iacovides and Dr. Tim Craft for their help and support during the period of my studies. Additionaly I wish to express my gratitude to Dr. Alex Skillen for being always available and willing to help with the Overset Grid Code. Of course all this would not have been possible without the economic resources provided by CONACYT, so I'm grateful to them as well.

I would also like to thank my colleagues Isidro Sanchez, Hector Guerrero, David Jimenez, Emmanuel Durazo and Leonardo Garcia for their valuable help. Particularly I want to express my gratitude to Petteri Kotovaara, Simone Marcenò, and Marcello Pagnan for their hospitality during my time in Europe and their friendship. Finally, I would like to thank my family back in Mexico for their continuous support and encouragement to complete this research.

Chapter 1

Introduction

1.1 Overview

Although fluid-solid interaction (FSI) phenomena play a significant role in most engineering applications, computer simulation techniques in this field are still at a developmental stage, and are constantly evolving. The strongly coupled nature of the physical process makes for a daunting and challenging modelling experience if accurate results are desired. As is the general trend in the computational simulation field, the increase in hardware performance has enabled the implementation of more computationally demanding coupling algorithms. Therefore nowadays simulation results are in principle capable of reflecting the characteristics of the problem better, but this does, of course, also depend on having accurate and reliable models for the physical processes incorporated in the software.

Within the field of Bio-Engineering, it is of special interest to understand the blood-artery interaction in order to propose new medical treatments for conditions such as aneurysms or stenosis. Civil engineering usually involves fluid-structure interaction analysis to ensure that buildings or bridges are able to withstand wind loads. Aerospace and turbomachinery industries are always on the lookout for lightweight and resilient materials which can cope well with extreme operating conditions due to their interaction with the surrounding fluid flow. There is consequently a strong case for the development of versatile FSI solvers which can deal with a wide range of problems. The main focus of this research is to contribute to the FSI modelling efforts pertaining to rigid-body fluid-solid interaction.

This work builds on a fluid flow solver previously developed by Skillen [1]. The base software solves the steady state incompressible Navier-Stokes equations in three physical dimensions, and is capable of modelling turbulence through the Reynolds-averaged Navier-Stokes equations (RANS). The finite volume method was selected to discretize the governing equations, and the physical domain can be discretized using overlapping grids. One of the main advantages of the overset-grid method is its ability to resolve critical zones, such as boundary layers in complex geometries, using structured grids. A direct consequence of using structured grids is the possibility to define grid connectivity information implicitly, thus avoiding the need to spend additional memory resources for such a task, as is needed in the case of using unstructured grids.

When overset grids are used there is the need to exchange information amongst grids. This inter-grid information transfer needs to take place using interpolation techniques. Nevertheless, the upshot of using any kind of interpolation is that conservation of the fundamental quantities is not preserved generally speaking. However, depending on grid sizing in the interpolation zone, and on the interpolation technique employed, perfectly valid solutions can be achieved. If standard bilinear or trilinear interpolation (SI) is selected, the user must be aware that in order to obtain satisfactory solutions fine grids in the interpolation zone are required. An alternative to standard interpolation is to use an interpolation algorithm based on the approximation of the mass conservation condition across grid interfaces. As its name suggests, mass flux based interpolation (MFBI) is the recommended choice for inter-grid communication, although it has been proven in [1] that at grid independence both algorithms perform satisfactorily. These interpolation choices are available for the user in the base software configuration.

As a starting point for this work there was a need to modify the grid input format for the computer code. Previously, the code read in grids generated using the proprietary application GAMBIT, which was part of the FLUENT package. Additional measures had to be taken by Skillen to ensure compatibility amongst the structured overset solver, and the unstructured (mesh connectivity) grids generated in GAMBIT. These measures included generating logically structured topologies, and an algorithm to transform how cells were

tagged natively by the employed grid generation tool, into an orderly cell tagging formulation. The main reason to alter this grid generation methodology was that the GAMBIT software is no longer supported by ANSYS, and it was therefore decided to build in the capability of using a non-proprietary grid format. The solution was to develop some code which takes grids written in the *CFD General Notation System* [2] (CGNS) as input, and whose output is a format suitable to be read by the existing overset code, and which can include boundary condition information as well. Since the CGNS system is a standard, there are several commercial or open source grid generators that can create grids in such a format.

The CGNS system is described as follows [2]: ' *The CFD General Notation System* (CGNS) provides a general, portable, and extensible standard for the storage and retrieval of computational fluid dynamics (CFD) analysis data. It consists of a collection of conventions, and free and open software implementing those conventions. It is self-descriptive, machine-independent, well-documented, and administered by an international steering committee. It is also an American Institute of Aeronautics and Astronautics (AIAA) recommended practice'.

The second step taken in this work towards an FSI implementation was to solve the laminar time-dependent incompressible Navier-Stokes equations. The two selected discretization schemes for the transient terms in the governing equations were: the Euler first order implicit scheme, and a second order implicit backward discretization scheme. The former was selected mainly for its stability and boundedness properties, whereas the latter was selected to provide the implementation with a more accurate choice whenever required.

The next logical step towards an FSI implementation was to formulate the fluid flow governing equations using either a Lagrangian, Eulerian, or hybrid flow description in an inertial reference frame to account for grid motion. This approach to handling moving grids is known in the literature as the arbitrary Lagrangian-Eulerian method (ALE). For incompressible flow in an inertial reference frame the method boils down to the computation of the relative mass flow rate through cell surfaces in the convective term of the conservation equations. Instead of calculating the necessary grid velocity to do so which

can be complex, a more general approach consists of separating the relative mass flow surface integrals into their absolute flow velocity and grid velocity contributions and solving them separately. This is convenient because another conservation equation needs to be satisfied when dealing with moving grids. Namely, the space conservation law ensures that grid motion (either pure translational motion, rotation, or grid deformation) shall not disturb the flow field. It turns out that for moving grids the space conservation law appears naturally within the mass conservation equation when using the ALE method, and one of the resulting additional terms is the previously mentioned closed surface integral of the dot product of the grid velocity with the surface area vector. This closed surface integral is then evaluated by acknowledging its physical significance thanks to the space conservation law. By computing such a term consistently it can be guaranteed that no spurious mass sources will appear in the mass conservation equation due to grid motion.

Finally, according to the desired FSI methodology, the next development step can be described in general terms as devising a FSI coupling strategy, and either implementing or using an existing solid solver. For this work a partitioned coupling approach was selected, and a three dimensional rigid-body motion solver was developed. The next section delves more into the two main different approaches of FSI modelling along with possible coupling strategies.

1.2 Fluid-Structure Interaction Methodologies

Traditionally the designer/analyst chooses to focus on either the fluid or solid aspect of the problem at hand and simplifies the other. Even though this approach might be useful in some situations, it is not acceptable for cases where the coupling is so intricate that simplifying either one yields incorrect results.

Considering what was mentioned above, several approaches are suitable for FSI modelling depending on the degree of accuracy required, and on the characteristics of the problem under consideration. The two main existing branches according to the behaviour of the solid body are: a) deformable solid body modelling, and b) rigid-body motion modelling. In the following subsections a brief description of the main methodologies employed for each of these approaches shall be discussed.

Regardless of the assumptions made about solid behaviour, the modelling engineer faces yet another equally crucial matter: the coupling strategy. Coupling can be achieved in a direct manner if both solid and fluid are solved concurrently. This approach however requires a powerful nonlinear numerical solver designed with this purpose in mind. Because the engineering field has historically been segregated regarding the simulation of solid and fluid problems, domain specific solvers are currently the norm rather than the exception. It is therefore more practical to integrate existing bespoke fluid or solid solvers together to tackle FSI problems, than to create one from the ground up. This is why a segregated coupling approach is far more popular.

Strong coupling can still be achieved with segregated solvers by means of outer iterations. If the interaction amongst the two media is not so intrinsic, one might even be able to obtain acceptable solutions with weak coupling techniques, and thus avoid the computational overhead inherent to segregated strong coupling algorithms.

1.2.1 Deformable-solid FSI modelling

If the problem requires the solid to be modelled as a deformable body, then the first obstacle for attacking numerically the coupled phenomena is the fact that both the solid and the fluid domains are usually discretized employing different approaches in most commercially available segregated solvers. While solid mechanics problems typically are dealt with using the finite element method (FEM), fluid mechanics problems are discretized via the finite volume method (FVM). Furthermore, the solution techniques employed by both methods to solve the discretized set of equations are radically different. Numerical fluid mechanics solutions are obtained with iterative methods, which are better suited to attack nonlinearities. On the other hand, solid mechanics problems typically rely on direct methods for such tasks [3].

With plenty of FEM and FVM commercial codes available nowadays, the traditional FSI approach is to set up the solid mechanics problem with imported stress boundary conditions. The imported boundary conditions result from a flow field solution obtained by using a different software package corresponding to the fluid with which the solid interacts. As discussed in [4] this procedure involves several factors that have to be considered

before committing to solve the problem in this fashion. The aforementioned boundary condition information has to be interpolated to the locations on the boundary where the discrete values of the dependent variables are stored in the FEM. This will unavoidably lead to errors that could be significant. Computational overhead is also likely to occur since the data storage structure of both approaches is quite different; this is exacerbated when dealing with transient problems where the transfer of information has to be completed for each time step.

An alternative to employing the two separate discretization methods is to solve both the fluid and solid domains using a single discretization technique. This, along with the same variable storage arrangement (i.e. staggered or collocated) for both domains eliminates the major issues mentioned above. The obvious questions then are: what are the limitations of the FEM when it comes to solving fluid dynamics problems? Is the FVM suitable for tackling solid mechanics problems?

With the ever increasing performance of hardware equipment the FEM has also overcome one of its major limitations when being considered for the solution of fluid dynamics problems, namely the usage of computationally expensive direct methods for solving the resulting system of discretized equations. In the same vein, one of the seminal contributors of the FEM, J.N. Reddy [5] acknowledges the fast pace at which the method has evolved and stresses that new solution algorithms are definitely improving its applicability to the field of fluid dynamics. The FVM has also earned a respectable reputation amongst researchers to attack solid dynamics problems, as will be seen in the literature review. The main reasons for this being the conservative properties of the FVM, and the inherent nature of iterative solvers to deal with nonlinearities.

1.2.2 Rigid-body FSI modelling

If the problem is suitable to model the solid as a rigid body then the solid governing equations are, in general, less challenging to solve. However, the coupling is still critical to obtain accurate solutions. In these cases the most challenging aspect to deal with is efficient fluid domain grid generation at each time step without degrading mesh quality due to solid body motion.

Fortunately there exist methodologies that allow efficient redefinition of the fluid computational domain at each time step, while preserving the original mesh quality. The immersed boundary method is an excellent possible choice for this purpose and it seems to be the most popular as well. The overset-grid method also falls under this category, with the added benefit of being particularly well-suited for problems requiring fine near-wall grids without compromising grid quality. Both approaches rely on somehow disabling or neglecting computational cells in regions occupied by the solid mass.

As for the solid-body motion solver, it is commonly based on high order numerical methods such as Runge-Kutta or Adams-Bashforth, since the solution of the Euler-Newton equations of rigid-body motion introduces virtually no overhead to the overall solution algorithm. For a robust rigid-body motion solver implementation, quaternions are without competition when it comes to populating the rotation matrix. Not only are they a reliable set of parameters that work for all possible orientations, but they also impose less computational stress on the simulation when compared to Euler angles.

Currently most commercial CFD packages offer rigid-body motion solver modules which can be coupled to the fluid flow solution. Generally speaking the user still has to develop a software interface to couple them. As of the time of writing of this work, at least one “black-box” commercial application has the capability of coupling the overset-grid method with a rigid-body motion solver. The potential offered by this pairing is such that two other major CFD distributions are in the process of implementing this methodology. Most notably, Pegasus 5, an overset rigid-body motion implementation which has been used by NASA for a long time now, has been the recipient of the software of the year award (2016) [6].

1.3 Objectives

The aim of the present work is to develop the capability to simulate one-way and two-way FSI phenomena with rigid moving boundaries. The integration of the overset-grid method, and a rigid-body motion solver, with a partitioned coupling algorithm is an appealing solution strategy for this complex problem. The overset-grid method offers tailored boundary fitted meshes overlapping a static background mesh, which is a combination naturally suited to handle moving bodies in a flow field in an efficient way.

Code verification and validation play a fundamental role in any numerical work. As such, another central objective of this work is to verify and validate this implementation. Verification of the main components of the solver is presented by means of the method of manufactured solutions (MMS). Validation is achieved through a set of cases with comparison to either experimental data where possible, or simulation results from the literature.

The impact of adaptive relaxation on the convergence rate of the partitioned coupling algorithm is presented. Weak and strong coupling algorithms are contrasted to determine the compromise needed between accuracy and computational efficiency to obtain satisfactory solutions.

1.4 Structure of this Thesis

Chapter 2 presents a robust literature review of the topics that are relevant to FSI simulations in general (i.e. the application of the FVM to solid mechanics problems, the overset-grid method, and previous FSI studies). Chapter 3 delves into the theory of the FVM, the overset-grid method, the ALE formulation of the governing equations, the space conservation law, the rigid-body motion solver, and the FSI coupling strategy. Solver verification procedures are presented in Chapter 4. Validation of the one-way and two-way model is shown in Chapters 5 and 6 respectively. Conclusions from this work are summarised in Chapter 7, together with suggestions for future work.

Chapter 2

Literature Review

2.1 Overview

The objective of this chapter is to present a review of the fundamental modelling techniques pertaining to this research, and to identify the advantages and limitations of using certain approaches for the undertaking at hand. Special emphasis has been placed on the overset-grid method, and on FSI simulations. Section 2.2 gives a brief insight on the main features of the FVM and its current development trends. Section 2.3 focuses on the use of the FVM to solve solid mechanics problems which is relevant for deformable body FSI simulations. Section 2.4 gives an overview of the evolution of the overset-grid method. Section 2.5 is a review of relevant work on FSI simulations, and Section 2.6 is dedicated to the conclusions of the literature review.

2.2 A brief overview of the Finite Volume Method

Since its appearance, the FVM has proved to be a compelling alternative approach to the finite difference method (FDM) for solving fluid dynamics problems. The FVM is specially well suited to deal with continuum mechanics problems numerically due to its conservative properties on both a global and a local scale [7]. Furthermore, the use of the finite difference method to properly resolve fluid flow over elaborated bodies implies cumbersome mesh transformation procedures to handle non-uniform grids [8]. On the

other hand the FVM offers a solution to attack complex geometry problems without significantly making the already painstaking process of mesh generation more complicated.

Perhaps the most serious limitation of the FVM is how difficult it might be to generate a higher than second order accurate formulation [9] . Most of the recent work in the development of this method has shifted towards formulating convection discretization schemes that preserve two highly desirable properties —boundedness and accuracy [10], or to pair the method with powerful domain discretization techniques such as overset grids, or the immersed boundary method. By combining the FVM with such powerful techniques its potential to solve elaborate problems is greatly enhanced, and it still remains the most widely-used approach to discretize the governing equations of fluid flow.

2.3 The FVM for Solving Elasticity Solid Mechanics Problems

Demirdžić et al. [11] as cited by Yates [4] were amongst the first researchers to employ the FVM to solve solid mechanics problems. Demirdžić and Martinović [12] applied the method to solve problems where the behaviour of the material was not limited to the elastic deformation range. They presented four challenging "thermo-elasto-plastic" cases in which the governing equations accounted for the conservation of thermal energy and momentum balance, complemented by general non-linear constitutive relations. In addition, an extra non linearity was introduced by allowing material properties to be a function of temperature. The implemented software allowed for unsteady and three dimensional problems to be tackled. After the set of discretized equations was linearised, a simple line-by-line tridiagonal matrix algorithm (TDMA) was used to solve the algebraic system of equations. The most challenging case presented was that of the arc welding process of a thin plate, which involved transient fully-elastic deformation of the material along with temperature-dependent properties. An energy source term was used to model the welding process, and the problem also included convective heat transfer at the boundary of the plate. Despite the low number of cells employed for the simulation (675) the results compared reasonably well with the available experimental data as seen in Figure 2.1, which

depicts temperature and stress profiles along a selected cross section of the plate.

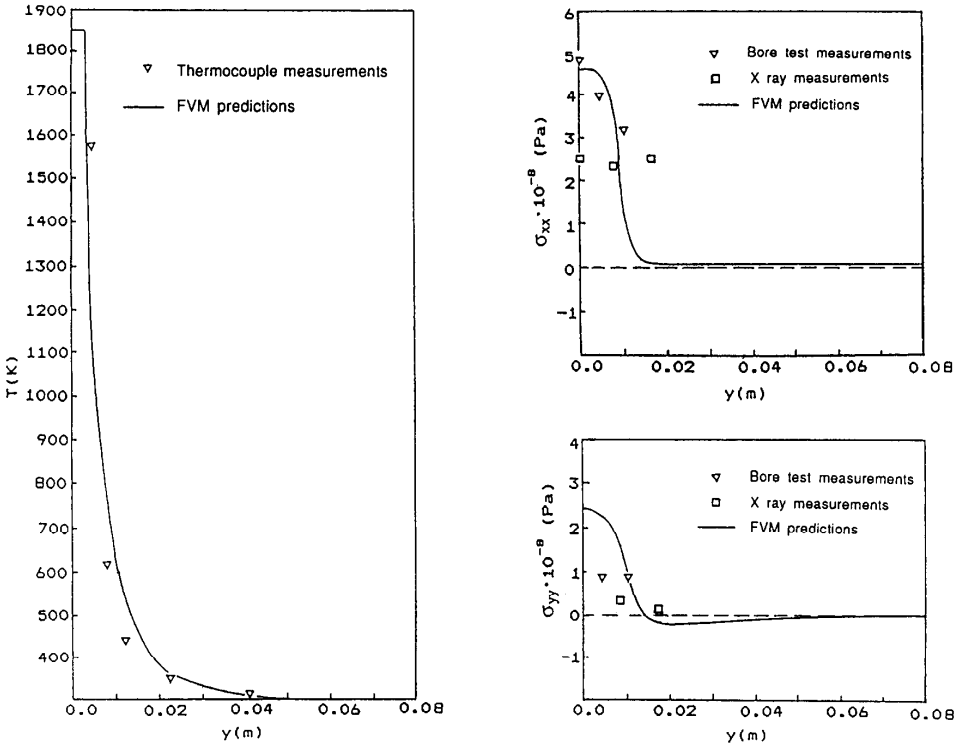

Figure 2.1: Temperature and residual stresses profiles. Image from [12]

Perhaps the only serious limitation of the above mentioned research studies was the fact that they both were constrained to use hexahedral computational cells. In a later project Demirdžić and Muzaferija [13] addressed this by introducing an unstructured solver to deal with intricate geometries. Through three validation cases the authors highlight the relative simplicity of the method, and the inexpensive computational cost of a segregated iterative solver. Though the cases are limited to linear constitutive equations, the employed conjugate gradient solver is well suited to handle any non-nonlinearities that may arise when dealing with solid materials exhibiting plastic behaviour. The discretized formulation is of second-order accuracy, and combined with the flexibility that unstructured meshes offer to locally refine critical sections of the domain, it achieves impressive results with modestly-sized grids.

Demirdžić et al. [14] introduced the usage of convergence accelerating techniques via the multigrid method. Though the main objectives of their study were to create a set of benchmark cases through which further developments in the FVM applied to solid

mechanics could be assessed, and to contribute towards improving the computational efficiency of the solver, the exposed results show evidence of the exceptional capabilities of the method to handle involved configurations by means of unstructured grids. Through the use of the Richardson extrapolation method [15] the error estimates were computed, taking advantage of the multiple existing solutions employed by the multigrid technique. The error estimates reveal startling accuracy which further emphasises the reliability of the FVM.

Once the capabilities of the FVM to deal with solid mechanics problems were demonstrated, Jasak and Weller [3] focused on optimising the discretization practice in order to accelerate the converge rate of the incomplete Cholesky conjugate gradient solver, which included a parallelised algorithm based on the "domain decomposition approach". Even though it did not make use of multigrid convergence accelerating techniques, the convergence rate was improved by a cautious choice of under-relaxation factors. Results also showed impressive scaling capabilities (see Table 2.1) which enabled them to address practical applications with meshes comprising more than 300000 cells, and to obtain solutions in time frames that were acceptable for market driven industry design processes.

No. of CPUs	CPU time ($5x10^{-5}$) (s)	CPU time (600 iter.) (s)	Speed up
1	35620.4	90785.2	1.00
2	22398.8	56605.2	1.60
4	11406.6	29244.2	3.10
8	4247.32	10218.6	8.88
16	3766.13	8498.09	10.68

Table 2.1: CPU time to convergence on a parallel machine. Table from [3]

The method can also be applied to dynamic solid mechanics [16], incompressible elastic [17] and hyperelastic [18] materials —where the incompressibility constraint is enforced via the conservation of mass governing equation. This results in a similar situation as when dealing with fluid mechanics –the inherent coupling of the system of equations is handled either by means of the SIMPLE [19] algorithm or by using the SIMPLEC [7] algorithm.

Recent trends in the development of iterative solvers for the FVM in the field of solid

mechanics have been to implement "Block-Coupled" solvers [20], and to speed up the existing segregated approach when dealing with highly nonlinear problems [21]. The former aims to improve the convergence rate of the solution, albeit at the cost of increased computer memory use. The novelty of the methodology lies in the fact that the resulting matrix is still sparse, but with a different structure when compared to the traditional segregated solvers, and the method retains an implicit approach which allows the strong coupling of the governing equations to be treated efficiently. The latter development trend improves the convergence rate of the solution by updating the nonlinear terms while the inner iterative process has not yet converged to the specified tolerance.

Development of the FVM to model specialised bioengineering applications ([22], [23], [24]) has also taken place recently. In [24], the application of the FVM to formulate a realistic model of the human hip joint was formally introduced with exceptional results. Finally, as presented in [23] , importing commonly used FEM techniques such as the penalty method to deal with contact stresses seems a plausible possibility which further enhances the applicability of the FVM in solid mechanics.

2.4 Overset Grids

The origin of overlapping grid techniques can be traced back to the early 1980s ([25],[26]) with applications to transonic potential flow. At the time, it arose as a result of the need to confront troublesome-geometry fluid-dynamics problems with the well established finite difference method, as proper definition of boundary surfaces for elaborate topologies proved to be an extra challenge for the already nontrivial grid generation procedure, since the standard finite difference numerical formulation does not allow for a direct way of dealing with nonuniform grids. This restriction implies significant transformation strategies with the purpose of recasting the governing equations and generating a uniformly spaced computational domain. Additionally, the use of overlapped grids is a compelling choice to overcome limitations imposed by interface matching block-structured grids. Such limitations include grid irregularities occurring at the corners of subdomains and at surface perimeter lines [27].

Atta and Vadyak [28] employed overlapping grids to model transonic potential flow

over an isolated wing, and over a wing-pylon-nacelle configuration (see Figure 2.2). The results for the flow over the single wing were then compared to results obtained by using a single grid for the computation of the flow field over the same wing configuration. As can be seen in Figure 2.3 the solutions show great agreement. Although the information transfer process between the overlapping grids took place via a second order Taylor series interpolation, the authors concluded it did not degrade the accuracy of the solution.

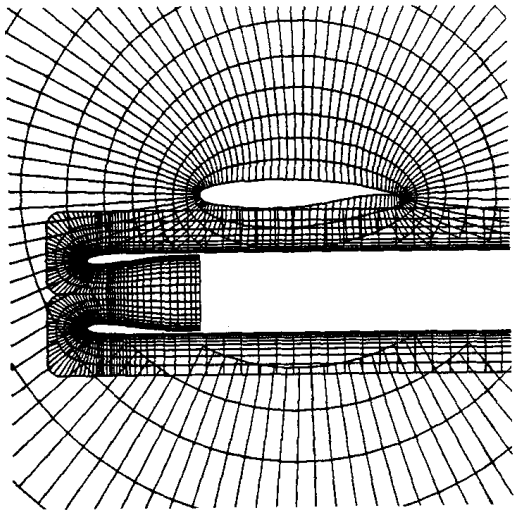

Figure 2.2: Embedded grids employed to model a wing-pylon-nacelle array. Image from [28]

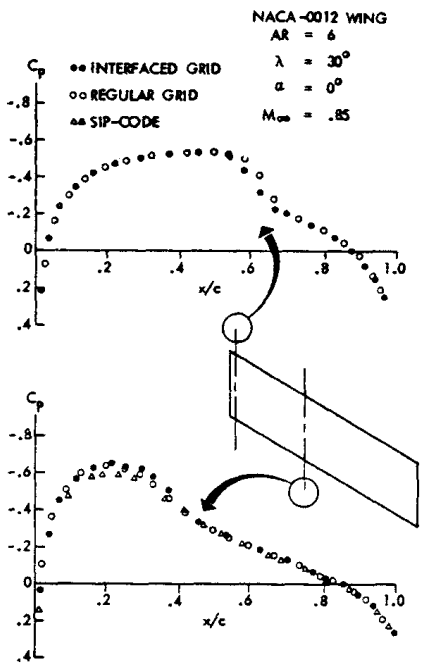

Figure 2.3: Comparison of Overlapped Grid Results With Single Grid Results. Image from [28]

A more complete treatment of the grid embedding technique was formulated by Benek

et al. [27]. Their work introduced a simplified hierarchical structure for the overlapping grids which allowed for the information transfer process between grids to be dealt with more efficiently and, equally important, in an ordered way. It also included two different and relatively simple alternatives to identify and tag correctly nodes that were not part of the computational domain belonging to the grid with the most primitive hierarchical identity, and by doing so they defined the concept of a "hole". In this framework they coined the term *Chimera method* to denote the applied grid embedding technique. With regard to the vital process of inter-grid communication, they used a first order accurate Taylor series formulation to interpolate the necessary data between overlapping grids.

Three cases were proposed in [27] to study the presented methodology, out of which the most interesting one is the analysis of the solution obtained for the flow field over a supercritical airfoil. Because of the shock wave features of the flow field for this problem, it provided for the opportunity to scrutinise how the weakest "link" of the methodology, the non-conservative first-order interpolation, affected the accuracy of the solution. Figure 2.4 shows the chimera grid used for the computation, and Figure 2.5 depicts the pressure coefficient on both the lower and upper surfaces of the airfoil. Three different numerical solutions are presented, of which two are single grid solutions (FLO43—potential flow and VAIR2D—Euler equations), and the Chimera grid solution—Euler equations. Experimental data is also shown in the figure to compare solutions. The large discrepancies are mostly attributed to the lack of viscous effects in the numerical models. On the other hand, the differences between the two numerical Euler equation models reveal the effects of the non-conservative inter-grid communication approach of the Chimera method. These effects are exacerbated by the violent discontinuities introduced by the shock wave.

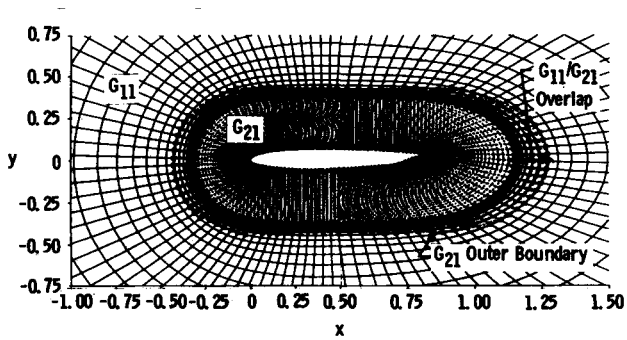

Figure 2.4: Embedded grids for the airfoil. Image from [27]

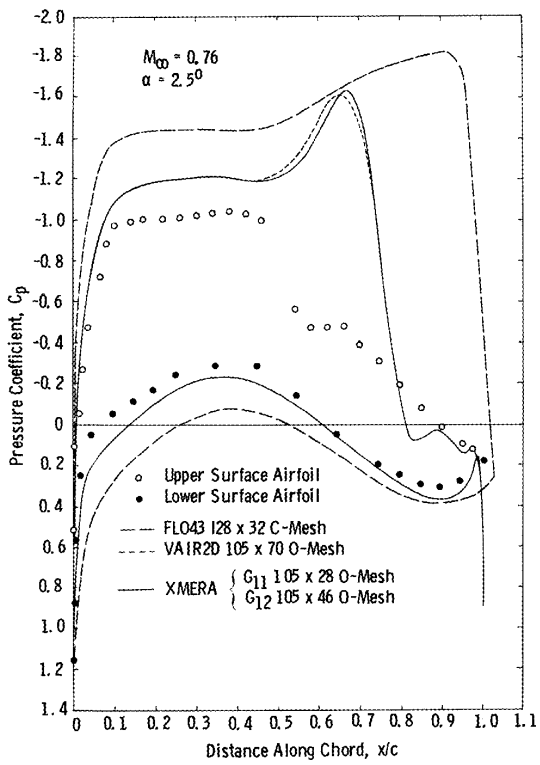

Figure 2.5: Solutions for the basic airfoil geometry. Image from [27]

One of the main difficulties when working with overlapped grids is how to handle inter-grid communication without degrading the accuracy of the solution due to errors introduced by the generally non-conservative interpolation procedure. Another vital part of any overset grid algorithm is how to carry out the difficult and computationally expensive task of determining the so called "hole" grid points. A relatively simple tactic is depicted in Figure 2.6 as presented in [29].

Initially one has to identify a surface which comprises points belonging to the secondary grid that will serve as the boundary of the hole. Next, the process of computing outward normal vectors to the boundary surface at each grid point that defines it is shown as item b) in Figure 2.6. Establishing an origin within the surface limited area has to be done in order to compute the greatest distance from it to any point on the boundary, which is then used to compute a circle whose radius is the aforementioned calculated distance. The first test is then performed on every grid point of the background mesh to determine if the magnitude of a position vector stemming from the origin of the hole, to the tested point, is greater or equal than the radius of the previously calculated circle; if it is, then the tested point lies outside the hole. If the magnitude of the vector is less than the radius of the circle, then the closest point on the boundary surface to the tested background grid

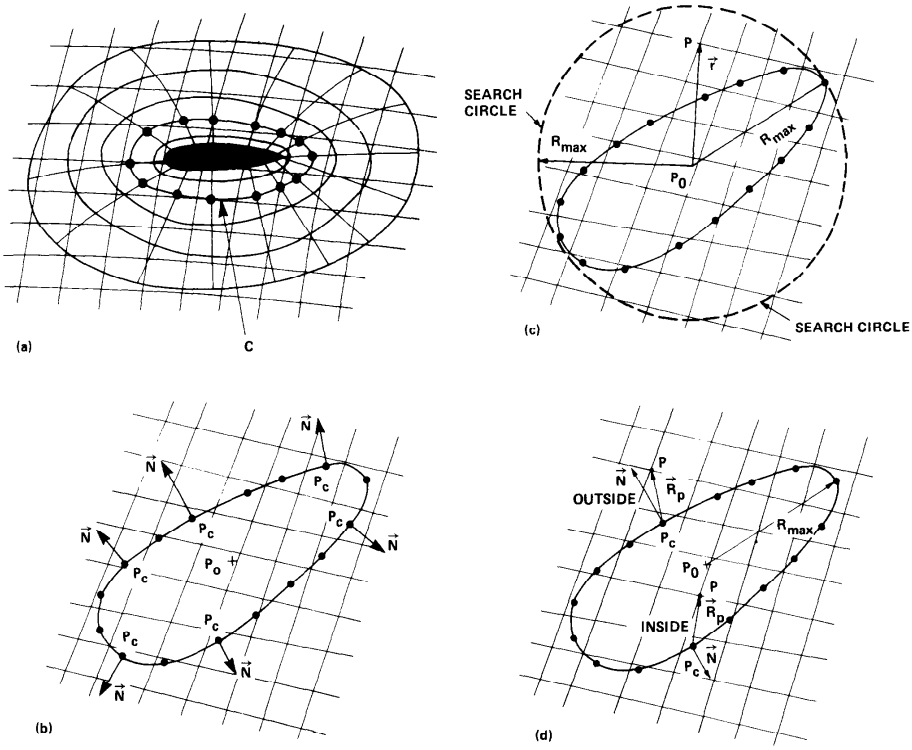

Figure 2.6: Hole cutting algorithm. Image from [29]

point has to be located, with the purpose of calculating the dot product of the outward normal vector on the surface point with the relative position vector stemming from the point on the surface to the point being tested. If the dot product is equal or greater than zero —then the point lies outside the hole.

So far all of the literature presented in this section has seen the application of overlapped grids to the finite difference discretization method. One of the first papers applying the methodology to the FVM is the one from Wang [30]; however, the highlight of his research is the implementation of a fully-conservative communication algorithm amongst overlapped grids. In recognition of the inherent extreme difficulties, specially in 3D, to implement the formally derived conservative constraints for overlapping grids, the basic concept is to *transform* the overlapping grid zone into a patched zone by means of a chosen boundary out of the ones available from the overlapping sections as seen in Figure 2.7

In this way the conservation requirement for the patched boundary, namely the flux of the chosen property leaving grid A-O is the flux entering grid B, is much easier to enforce (its implementation will be discussed in more detail when it is compared to semi-conservative approaches). Although this approach requires making changes to the primary grid due to the need to generate a new interface zone, the resulting communication

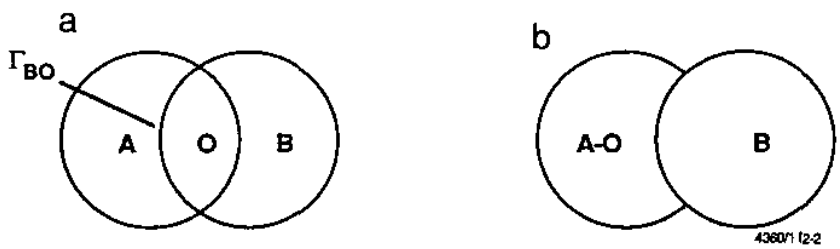

Figure 2.7: Transformation of overlapped zones to patches zones. Image from [30]

between grids is fully conservative, has only one solution for the originally overlapped zone, and most importantly it eliminates the need for interpolations. Four 2D cases were set up to compare results with the original Chimera non-conservative method of [30] — eventually the original Chimera code evolved from using the finite difference method to using the FVM, and it was renamed as "CFL3D".

Table 2.2 shows the mass imbalance computed for transonic flow through a channel with a 10% bump using three different codes. A significant reduction in the order of magnitude of the mass imbalance can be achieved when compared to the code using data transfer via non-conservative interpolation.

	Inflow mass rate	Outflow mass rate	Imbalance
CFD-FASTRAN single code	269.52534	269.52535	$4x10^{-6}\%$
CFD-FASTRAN conservative	269.47414	269.47412	$6x10^{-6}\%$
CFL3D non-conservative	269.8248	270.8508	0.38%

Table 2.2: Mass Flow Imbalance. Table from [30]

Discretization of the physical domain with overlapped grids is definitely an enticing choice. However, the real superlative advantage of the method over other strategies is seen when dealing with fluid flow over moving bodies. In such cases the spectrum of reasonable choices for generating boundary-conforming grids is limited to three methodologies [31]:

- *Grid Deformation Approach*
- *Grid Remeshing*
- *Overlapping Grids*

The grid deformation approach allows for the computational domain to be "adjusted"

at each time step. Since this methodology retains the original grid topology and the total number of control volumes, the motion range of the moving body must be limited to small translations, otherwise the mesh would be severely distorted. The advantage of the method is that the flow solver retains its conservative properties given that no interpolations are necessary.

As its name suggests, grid remeshing involves a complete near body mesh generation for every time step in order to account for the current position of the body. While this technique is flexible with regard to the motion range of the moving body, it is necessary to interpolate flow information from the old mesh to the current mesh —which obliterates the local conservative properties of the FVM. The need to generate a new mesh at every time step could also result in computational overhead.

Overlapping grids offer the capability of having a body-fixed secondary mesh moving relative to the main or background mesh, which eliminates any adjusting or remeshing needed to account for the motion of the body. Although ultimately there will still be the need to transfer information in the overlapping zone, this can be achieved in semi-conservative or conservative manners.

In recognition of the above alternatives, Wang [32] extended his previous work [30] to account for flow over moving bodies. The moving body condition presents two new challenges to the problem: The moving mesh must not introduce any disturbances to the flow —which is handled by means of the "*Geometric Conservation Law*" (GCL, also known as the Space Conservation Law), and the emergence / vanishing of cells in the computational domain due to the movement of the body fitted grid must be treated in a consistent and systematic way to allow for the governing equations to be applied properly.

As seen in Figure 2.8, at time step n cell B has to be neglected from the major grid for computation purposes, however; the same cell B has to properly be considered by the major grid at time step $n+1$. A similar situation is experienced by cell A, the difference being that it is a "cut" cell at time step n, and a hole cell at time step $n+1$. Under these circumstances the discretized governing equations cannot be solved for such cells since they require valid scalar values for at least one previous instant in time. In order to address this issue the authors employed a cell-merging technique (combining individual cells to

form a bigger control volume) along with an imposed condition on mesh movement. This implies limiting grid motion in such a way that the generated patch boundary between overlapping zones does not cross any corner vertex of a merged cell between consecutive time steps, so that the topology of a merged cell remains constant during one grid motion step.

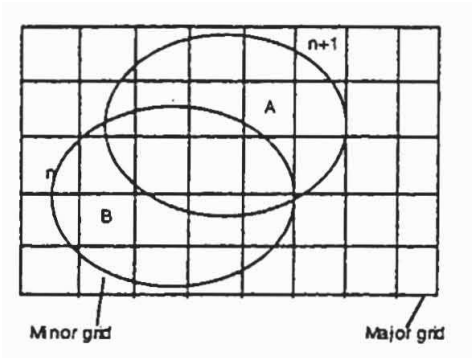

Figure 2.8: Vanishing and new born cells . Image from [32]

For moving grids, the general conservation equation includes a contribution from the grid velocity in the convective term. In order for the FVM to keep its conservation properties in this case, the Geometric Conservation Law must be satisfied. The Geometric Conservation Law states the following [9]: "The sum of volume fluxes through control volume faces due to their movement must equal the rate of change of volume". One has to make sure the GCL is satisfied in the discrete sense, otherwise any free stream will be disturbed by the moving mesh. Two store separation problems were presented, out of which the most impressive one was the 2D supersonic flow modelling of a space vehicle releasing a missile. Figure 2.9 shows the overset grid for the problem.

The main drawback of this alternative method is the fact that the transformation process from the overlapped grid to the patched boundary in order to achieve a fully-conservative interface treatment has to be done for every time step. At the time, Wang estimated this process accounted for about 30% of the total calculation time of the simulation.

More recently, Tang et al. [33] proposed a novel treatment for the transfer of data in the overlapping zone of embedded grids. Their proposition is a compromise between the accuracy and local conservation properties offered by a fully-conservative methodology, and the inherent simplicity offered by an interpolation scheme. In recognition of the

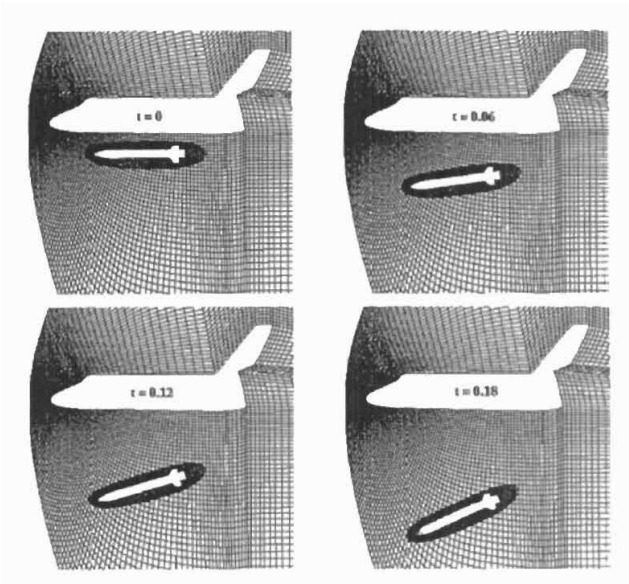

Figure 2.9: Computational grid at different times. Image from [32]

potential computational overhead and difficulty of implementing a complete local conservative conceptualisation which involves patching the boundaries of overlapped grids, the authors rely on satisfying mass conservation on a global scope.

The data transfer process focuses exclusively on generating a second-order accurate approximation of the required mass conservation condition in the composite domain — namely the major grid calculated mass flux crossing the common interface must match the minor grid calculated mass flux crossing the same interface. Acknowledging the fact that strictly speaking the proposed method thoroughly satisfies the mass conservation condition only in the limit of zero grid spacing, they decided to name it *Mass Flux Based Interpolation* (MFBI).

Comparing the MFBI with its fully-conservative counterpart, the one proposed in [30], the main difference is the way the fluxes over the interface are computed for the major grid, see Figure 2.10. Whilst in the fully-conservative implementation the interface flux for the major grid is computed by considering all the contributions from the small faces formed by the intersection of the major grid with the boundary surface, in the MFBI approach this is done by interpolating the velocity components from the cell centre of the major grid to the nodes of the minor grid that intersect the interface boundary, and using those values to calculate the mass flux from the major grid.

A comparison between the MFBI and standard trilinear interpolation (SI) revealed that

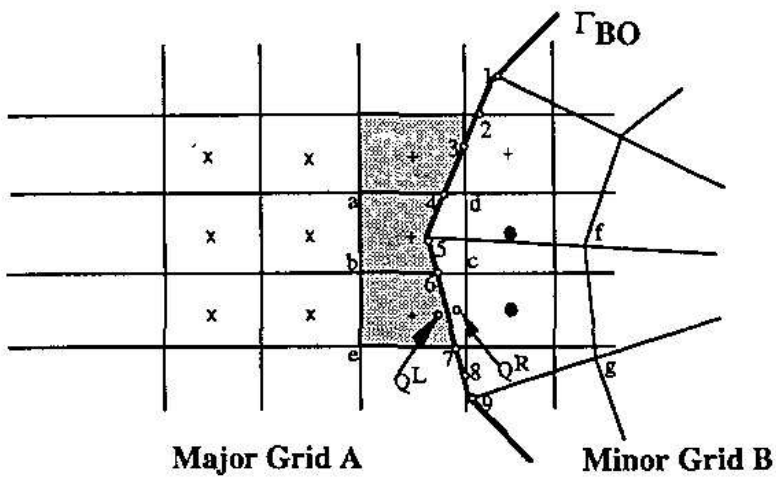

Figure 2.10: Schematic of a conservative treatment. Image from [30]

SI tends to produce nonphysical pressure fields, and severely hinders convergence rates. Interestingly, the authors of [30] conclude that the use of SI overspeficifies boundary conditions for the discrete problem at the internal interface, which potentially could lead to an ill-posed problem.

The use of structured-overlapping grids emerged to satisfy the need of tackling complex geometry problems with the finite difference discretization technique. On the other hand, the introduction of the finite volume discretization method made it possible to discretize the governing equations in arbitrarily shaped control volumes. With this capability it became much easier to generate grids that conformed properly to arbitrary boundaries and led to the development of unstructured FVM solvers. As seen so far, eventually the embedded grid methodology was implemented successfully along with the FVM. With this in mind, nowadays when dealing with topologies requiring special attention, there are two separate choices an FVM analyst can make. Considering the fact that almost all (if not actually all) state-of-the-art commercial FVM solvers are able to handle unstructured grids some interesting questions arise: Has the overset-grid method ever been used alongside unstructured grids? If so, what are the advantages of doing it?.

Aside from the fact that it is relatively easy to generate boundary fitted unstructured meshes (although high quality meshes are not so simple to generate), the process of grid adaptation is significantly more flexible for unstructured grids. As its name suggests, unstructured grids are suitable for local grid refinement without issues, whereas a structured

grid does not allow for such treatment since the arbitrary introduction of nodes in a local zone would destroy the global structure of the mesh. This forces the whole grid to be adapted which might not be optimal and demands additional computational resources. In addition to this, the grid adaptation process comes with some extra perks such as being able to estimate the discretization error of the solution, and introducing convergence accelerating methods such as the multigrid technique. Therefore, by combining the benefits of overlapping grids with unstructured meshes, some really interesting options arise, especially when modelling deforming or moving bodies.

Wang and Kannan [34] and more recently Wang et al. [35] have exploited this appealing combination and have been able to solve rather interesting fluid flow problems. Figure 2.11 shows an unstructured overset adaptive Cartesian-prism mesh from [34].

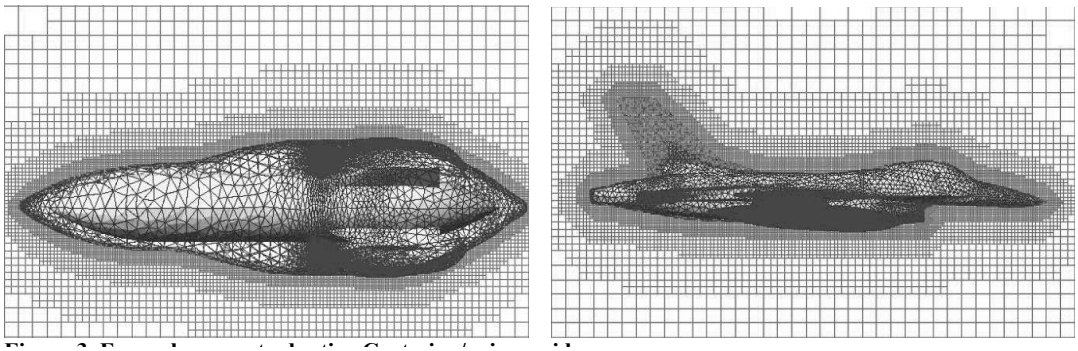

Figure 2.11: Unstructured overset Cartesian-prism mesh of a fighter jet. Image from [34]

In the case of [35] the authors took advantage of the exceptionally good capabilities of the overset method to handle moving bodies and solved turbomachinery problems using Large Eddy Simulations (LES). Figure 2.12 shows a zoom of the coupling mesh for a rotor-stator interface.

Skillen [1] implemented a steady state structured three dimensional overset-grid method solver for incompressible flow. The study focused on the implementation of an efficient algorithm to improve the performance of the hole cutting task. Furthermore, a new digital tree structure was developed as well as an assessment of two different ways of providing the means for inter-grid communication. For validation purposes some cases were compared to experimental data, as well as comparison with relevant commercial CFD tools.

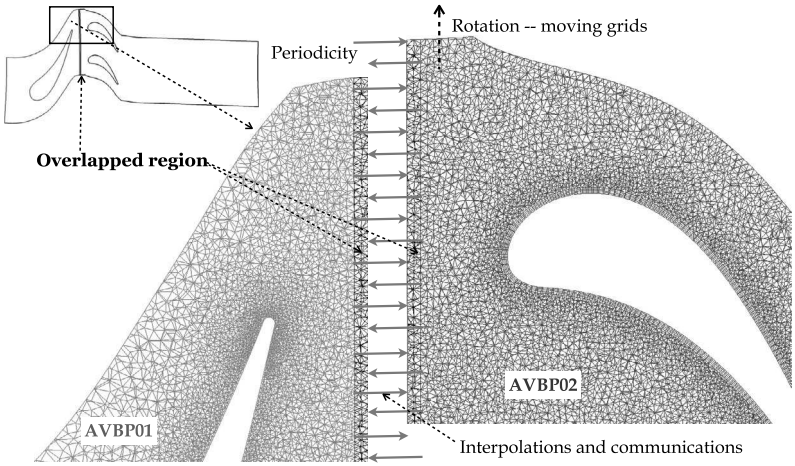

Figure 2.12: Unstructured overset Cartesian-prism mesh of a rotor. Image from [35]

Another main objective of Skillen's work was to assess the behaviour of a bilinear/trilinear interpolation algorithm used to transfer information between grids. The comparison with the semi-conservative MFBI described above served to clearly establish the consequences of using a non-conservative approach in terms of solution accuracy and convergence rates. Skillen reported the standard interpolation algorithm performed satisfactorily when grid independence was achieved. However, even in the cases where there was not a great difference amongst the semi-conservative and standard trilinear/bilinear information transfer, the swift convergence rate of semi-conservative interpolation on its own makes it worthwhile to use in every case.

One particularly interesting case presented by Skillen was that of a three dimensional carotid bifurcation. This artery is crucial to the cardiovascular system of human beings. With the emergence of fluid-solid interaction modelling, there has been a particular interest in physiological flows which could help understand, and ultimately propose novel treatments for life threatening conditions such as stenosis. Figure 2.13 depicts the overset mesh used for this problem, which comprises five sub-grids.

Laminar flow with three different Reynolds numbers was modelled along with the MFBI semi-conservative approach for inter grid communication. As shown in Figure 2.14 results compared well with the available experimental data, and most of the discrepancies were attributed to the inherent differences in the geometry of the model with the real artery.

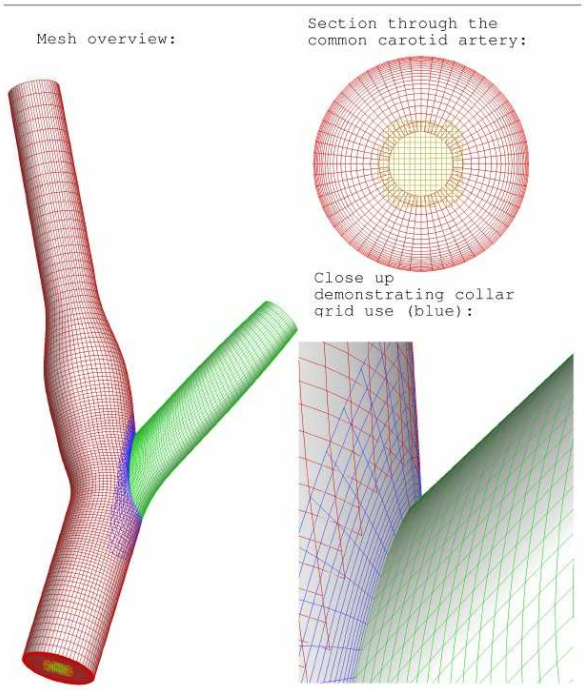

Figure 2.13: Overset mesh for carotid bifurcation.Image from [1]

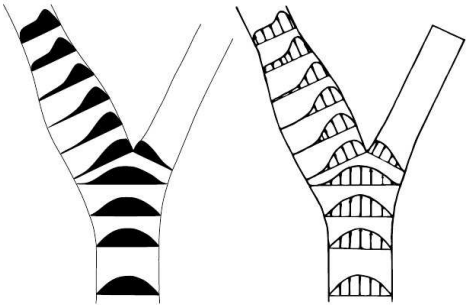

Figure 2.14: Velocity vector in the symmetry plane for the carotid bifurcation. Computed results (left) and experimental data (right). Image from [1]

2.5 Fluid-Solid Interaction Simulations

Fluid-solid interaction (FSI) phenomena is complicated by nature since it involves coupled behaviour. In this context the word *coupled* refers to the intrinsic effect either the solid or fluid medium has on the resulting reaction of its interface split complement. The extent to which the problem at hand is coupled can vary significantly, and in some cases it may be either simplified or directly neglected. It is one of the main objectives of the present study to be able to capture this coupled behaviour in such a way that it does not compromise the accuracy of the solution, hence no assumptions are to be made in the interface between the solid and the fluid.

There are two general coupling techniques to solve a FSI problem:

- *Monolithic approach*
- *Partitioned approach*

In a monolithic implementation the entire set of non-linear algebraic equations arising from the discretized governing equations of fluid flow and solid mechanics is solved as a whole [36]. That way, there is no need to treat the interface section separately in order to couple the problem. Since the assembled matrix of discretized equations is non-linear, it is usually solved by the powerful Newton-Rapshon method or some variant of it [37]. This strategy is generally expensive in terms of computational resources and requires some special preconditioning of the system of equations to be solved. Having mentioned that, because its solution is based on the Newton method it usually exhibits a better convergence rate than its partitioned counterpart ([38], [37]). Due to the deep-rooted interaction between the fluid and the solid while formulating and solving FSI problems with this technique, it is well suited to handle strongly coupled behaviours [39].

In contrast, the partitioned technique relies on solving the fluid and solid domains in a segregated manner. This allows for either the same or different equation discretization method to be used for the two domains. The interaction of both domains is handled at the interface and generally there are three ways of handling the coupling [36]:

- *Unified mathematical model*

- *Unified discretization method and boundary coupling consistency*
- *Unified solution procedure*

The unified mathematical model involves solving analytically a simplified form of the governing equations either on the solid domain or in the fluid domain, and then adapting the solution so it can serve as a boundary condition for the other domain. Greenshields et al. [40] employed this concept to model the fluid-structure interaction in plastic pipes. The validity of this method depends on the purpose of the simulation as well as on how close the simplified mathematical model resembles the actual physical situation.

Unified discretization and boundary coupling consistency interface treatment consists of solving numerically either the fluid or solid domains, or both, and applying the resulting pressure or displacement fields as boundary conditions for the respective medium. In the partitioned approach context, this is the most widely applied methodology at the interface to couple the problem. However, a key factor that directly impacts the accuracy of the solution is how to handle the transfer of information from one domain to the other — *if* the locations where the dependent variables are stored in each region means that interpolation is required, then the accuracy of the overall model can depend heavily on the interpolation scheme.

The unified solution procedure calls for the system of discretized equations to be solved monolithically as described previously. Since the scope of the present study is to solve the problem in a partitioned manner, most of the reviewed published work in the next paragraphs will be devoted to handling the coupling with the unified discretization method.

Traditionally the governing equations of fluid flow are derived in an Eulerian reference frame by means of the material derivative —since in most applications following a single fluid particle is of no particular interest. On the other hand the governing equations of solid mechanics are usually presented in a Lagrangian reference frame. When solving fluid-structure interaction problems with a partitioned solver it is of vital importance to account for the changes in the fluid physical domain due to the deformation or rigid movement of the solid body, these changes originate at the fluid-solid interface. As is often the case in user generated grids for numerical fluid flow problems, the mesh also conforms to

an Eulerian reference frame. In this fashion it is straightforward to handle large "deformations" of the flow field without issues —since the mesh nodes do not follow material particles of the physical domain.

As one would come to expect, an analogue situation takes place in numerical solid mechanics. The computational domain adopts the same reference frame as the governing equations, in this case being a Lagrangian reference frame, and a given computational node is associated with a material point of the physical domain. This way, tracking free surfaces or interfaces between different materials becomes much easier. The drawback of this computational domain methodology comes when large deformations of the material arise, causing the computational domain to experience such large deformations as well which might break the underlying solver. Figure 2.15 exemplifies the Lagrangian and Eulerian computational approaches.

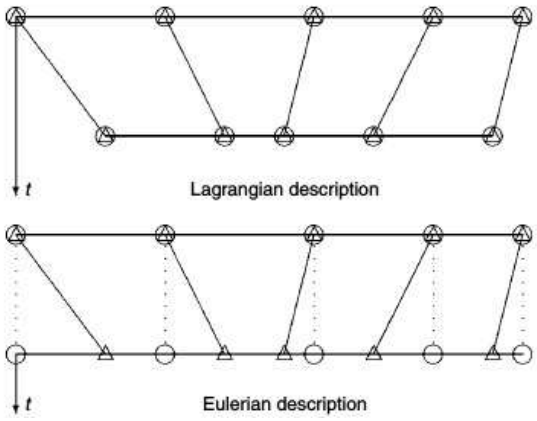

Figure 2.15: Example of Lagrangian and Eulerian meshes. Image from [41]

The Arbitrary Lagrangian-Eulerian method (ALE) implements a combination of both the Lagrangian and Eulerian descriptions to take advantage of their strengths. The result is a powerful method to solve problems in which both large deformations and tracking of surfaces or interfaces matter. Figure 2.16 shows a comparison between the ALE method and the Lagrangian approach for a finite element transient simulation to model the detonation of an explosive charge in a cylindrical vessel partially filled with water [41]. It can be appreciated how on the Lagrangian mesh after detonation, the grid is severely distorted, whereas the ALE computational domain retains the interface definition and copes well with the extreme deformation.

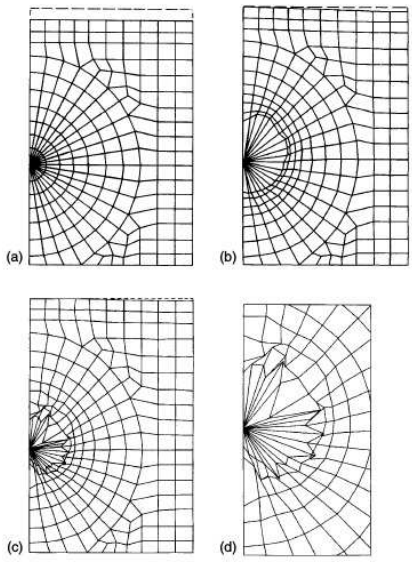

Figure 2.16: Lagrangian vs ALE descriptions: a) Initial FE mesh; b) ALE mesh at t=1ms; c) Lagrangian mesh at t=1ms; d) Details of interface in the Lagrangian description. Image from [41]

2.5.1 Instabilities in FSI partitioned algorithms

Even when both solid and fluid domains are discretized in time using implicit schemes, the partitioned information transfer process introduces an explicit element into the coupling routine. Hence, all partitioned FSI algorithms exhibit inherent unstable behaviour under certain conditions. This phenomenon is generally refered to as the *added-mass* effect [42]. A wide range of factors are responsible for triggering instabilities and they are thoroughly explored in [43] including: high fluid-to-solid mass ratios, fluid viscosity and structural stiffness, geometrical nonlinearities, order of accuracy of the time discretization schemes for both fluid and structural predictor. To make matters more complicated, if the simulation is on the onset of those instabilities a common sense countermeasure such as decreasing the time step size may actually worsen the condition as demonstrated in Figure 2.17, which is rather counterintuitive.

Analogously, selecting a higher order time discretization scheme may further aggravate the instability if the simulation is already presenting oscilating characteristics. The authors [43] present several useful results which serve to make the reader aware of the added mass effect condition.

Having reviewed the basic concepts of FSI simulations, it is now appropriate to take a look at relevant studies of the two main branches of FSI problems according to the

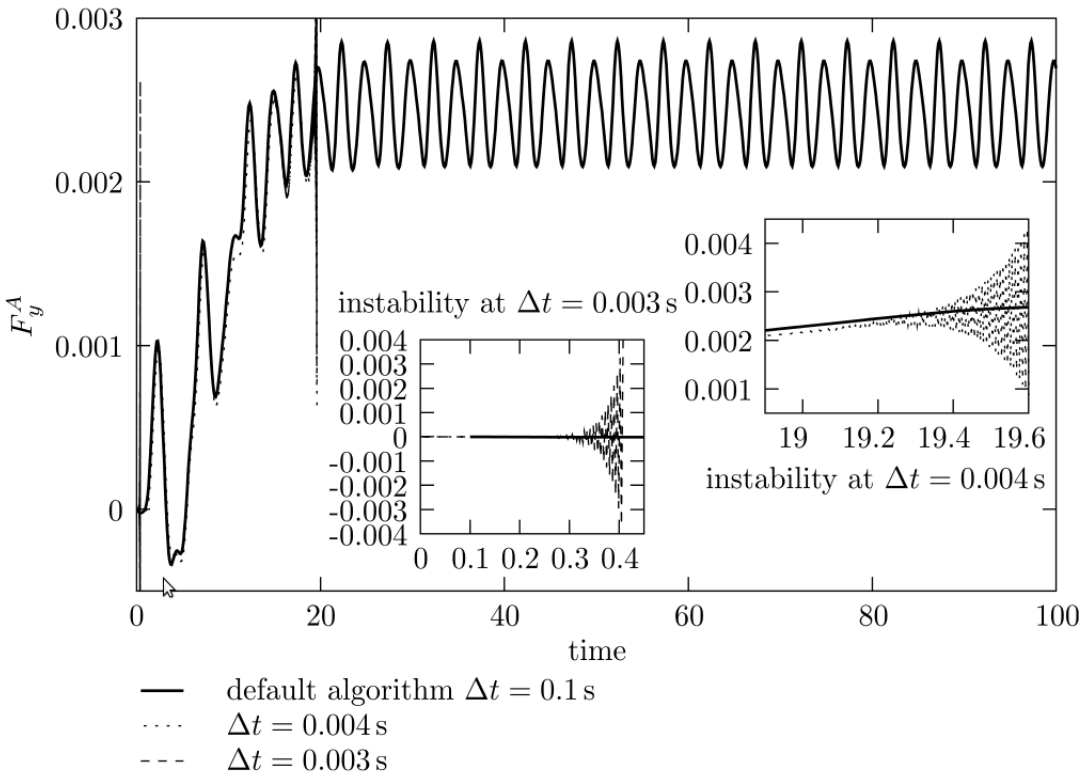

Figure 2.17: Evolution of the interfacial coupling force evaluated with different time step sizes. Image from [43]

methodologies defined for solid body modelling in 1.2.

2.5.2 Elasticity FSI problems

The work of Farhat et al. [44] can certainly be regarded as revolutionary in the field of FSI. They implemented a partitioned FEM-FVM solver using the ALE method for the transient movement of the fluid computational domain with application to aeroelastic problems. At the time, one of the novelties of their work was what they defined as a staggered explicit-implicit time integrator in which the transient response of the solid domain was advanced implicitly, while the fluid time advancement was done explicitly.

The reason for this approach was twofold; there was an existing Navier-Stokes explicit solver, and the structural response of the aerolastic structure is best suited for implicit time advancement due to the low frequency dynamics that dominate the motion. Another interesting concept introduced was that of a three-field coupled problem —where the moving fluid mesh can be considered to be a “pseudo-structural” entity. For this three-field coupled system an algorithm called *ALG0* was proposed which comprised the following

steps:

1. Initially perturb the structural entity.
2. Update the "pseudo-structural" entity to match the new structural boundary.
3. Advance the flow solution with the updated boundary conditions.
4. Compute the structural solution with the new flow induced boundary conditions.
5. Go back to step 2 until the full transient solution is obtained.

Another early example of a partitioned FVM-FEM study is the work of Glück et al. [45] to solve for lightweight membrane structures interacting with fluid flow. The coupling depended on the transfer of information between the two non-matching grids, and was handled by a conservative algorithm scheme pioneered by Farhat et al. [46]. To account for the mesh motion the ALE technique introduced by [47], and reviewed above, was used.

As seen so far, the use of a partitioned procedure along with different numerical methods to obtain the solution for each continuum unavoidably places an additional burden in the process of transferring information between both computational domains.

A novel concept to FSI modelling was presented by Slone et al. [48]. The authors implemented a single governing equation discretization method —the FVM for both media. Furthermore, by using a vertex based FVM for the solid continuum, the transfer of information between both media was done efficiently and without the need to employ non-conservative interpolation methods. Figure2.18 presents the different locations to store the dependent variables when using finite element (FE), cell centred FVM and vertex centred FVM. With this clever dependent variable storage location scheme for both domains, the need to interpolate displacements from the inner solid domain to the fluid-solid interface is avoided.

Bio-engineering or Biomedical applications are particularly suited to the application of FSI analysis. De Hart et al. [49] formulated a three dimensional FSI problem to analyse human aortic valves. Kanyanta et al. [50] rigorously validated their FSI model for flow in arteries with experimental and analytical data. Applications in nanotechnology have also

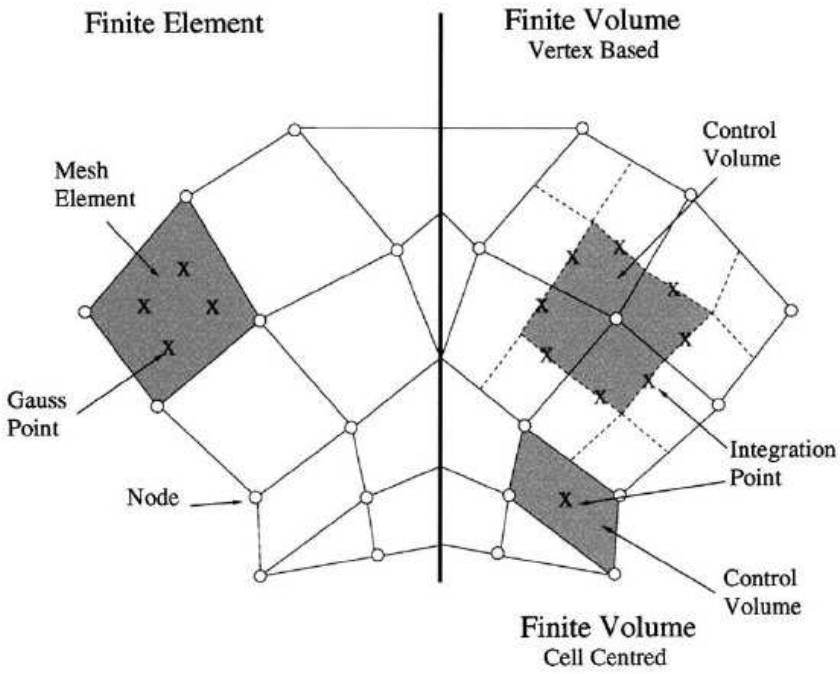

Figure 2.18: FE-FV discretization. Image from [48]

arisen as shown in[51]. As seen so far, the applications of FSI modelling are near endless, and there are constantly efforts to improve existing methodologies.

Another elasticity FSI study is that of Yates [4]. His work focused on physiological FSI problems both transient and steady state. Yates' implementation of elastic FSI modelling is a combination of some interesting concepts introduced earlier in this section such as: Structured non-orthogonal partitioned solver, ALE method for transient cases, a three-field approach, and a single governing equation discretization technique for both domains, using cell and vertex centred FVM formulation for the fluid and solid domains respectively.

Through the use of the RANS equations and the Launder-Sharma low-Reynolds-number model [52] early transition turbulent flows arising in physiological applications were able to be simulated. A common affecting medical condition known as *stenosis* is caused when a tubular shaped organ experiences a reduction of its cross-section [53]. The developed constriction can have a severe impact on the circulatory system and therefore it is important to understand the implications of a stenosed artery. Figure 2.19 shows the physical domain whereas Figure 2.20 shows a zoom of the generated mesh for the steady state simulation of flow through a compliant stenosis.

Simulations were carried out for different Reynolds numbers, even though a converged

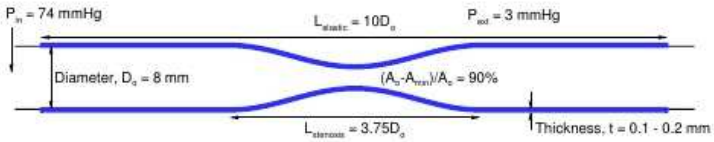

Figure 2.19: Compliant wall, stenosed geometry. Image from [4]

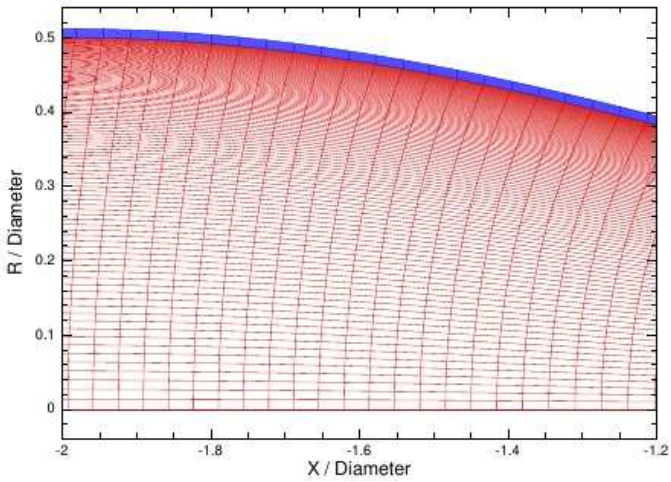

Figure 2.20: Computational domain detail of the compliant wall. Image from [4]

solution was not always possible. Yates attributes this to difficulties in resolving the transition to turbulent flow. Figure 2.21 compares the calculated pressure drop across the stenosis to experimental data. Data shows good agreement up to a point when the tube might have been deformed in a non-symmetric cross-section.

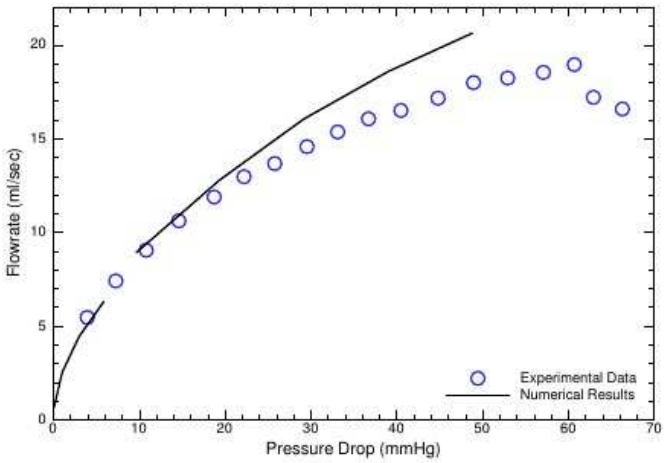

Figure 2.21: Flow rate comparison through compliant stenosis. Image from [4]

Finally, Yates presented the case of unsteady turbulent flow through a compliant aneurysm. An aneurysm could be considered to be the opposite of an stenosis —a bulge formed in an artery or vein. Figures 2.22 and 2.23 respectively show the computational

domain and a sample of the obtained results when they were compared to another numerical study respectively. Figure 2.23 presents a comparison of the velocity at the exit section of the aneurysm. The author considered an underprediction of the wall displacement to be the major cause of the disparity in the obtained results.

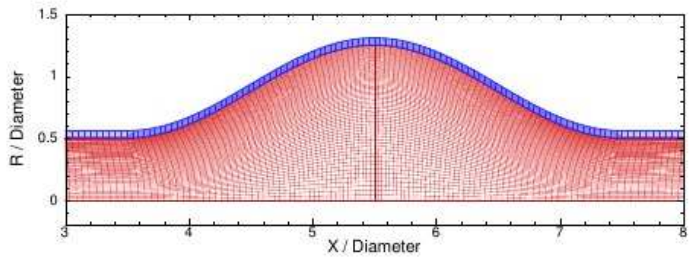

Figure 2.22: Computational domain of the aneurysm section. Image from [4]

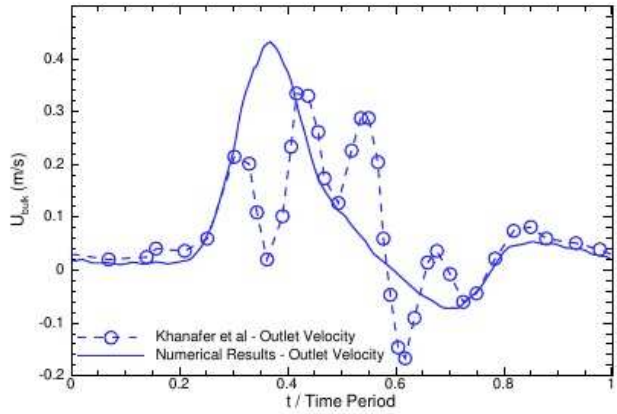

Figure 2.23: Comparison of the transient aneurysm exit velocity. Image from [4]

In conclusion, Yates' work succeeded when compared to the available numerical and experimental data in the literature for all the cases he studied, except for the one presented above. Yates acknowledged some underprediction in the displacement of the solid, which was perhaps caused by an incorrectly specified external pressure. Finally he recommended the extension of his work to a three dimensional simulation environment.

A more general approach to elastic fluid-structure interaction using the overset-grid method can be found in [54], [55], [56], and [57]. In particular the study of [57] employs a FEM-FVM partitioned solver along with the overset-grid method. Results show a significant improvement over the traditional ALE method with regard to mesh quality and the avoidance of having to re-mesh the whole fluid computational domain for moving bodies.

2.5.3 Rigid-Body FSI problems

The final section of this literature review pertains to the application of the overset-grid method, or other non-standard domain discretization techniques paired with a finite volume formulation for rigid-body fluid-structure interaction analysis.

As briefly discussed in 1.2.2, if solid deformation is negligible, the effects of any interaction with fluids are captured through the governing equations of rigid-body dynamics. This set of ordinary differential equations is classified in two groups where body acceleration is the link amongst them. Kinetics is modelled through the Euler-Newton equations which establish a clear connection between motion and its causes. On the other hand, the kinematic governing equations provide a detailed description of body movement at any given instant in time.

Out of all the numerical methods available to solve ordinary differential equations, by far the most widely employed to solve rigid-body FSI problems are the ones belonging to the broad predictor-corrector family. These methods are preferred because their algorithms blend perfectly well with the iterative nature of partitioned FSI solvers. There are also plenty of high-order Runge-Kutta methods. Runge-Kutta methods are based on the evaluation of the forcing term in the governing differential equation at different intervals within a single time step. This approach is similar to a standard predictor-corrector technique but the literature regards them as a different class [58]. A potential disadvantage of Runge-Kutta solvers is the requirement to evaluate the previously mentioned forcing term several times per time step. Depending on the degree of FSI coupling selected for the partitioned solver, this could represent an unnecessary burden on the general solution algorithm. Whereas Runge-Kutta solvers rely on information at different sub-interval points, multistep methods rely on data at previous time steps. These data include the costly computation of the forcing term in the differential equation so it is readily available. Although in theory multistep methods do not require a large number of forcing term evaluation, in practice, when combined with a strongly-coupled partitioned algorithm for FSI simulations, the outer coupling iterations could inevitably lead to computational loads comparable to the ones found in Runge-Kutta solvers. For this reason it is equally common to find FSI simulations using either approach in the literature [59].

Frequently, the actual range of motion of rigid bodies exceeds by a large margin that of elastically deforming entities. As such, out of the possibilities presented in 2.4 to discretize the physical domain every time step, the grid adaptation strategy can be practically discarded due to potential large mesh deformation.

A rather popular alternative method to the ones so far presented is the Immersed Boundary Method (IMB). In terms of grid generation efficiency this is perhaps the simplest choice, as in most of its variants a static non-conformal Cartesian grid is used to discretize the fluid physical domain, whilst the body generated mesh comprises a set of Lagrangian nodes. The concept behind the method is to account for solid interaction on the fluid side by means of the addition of source terms that represent wall forces in the fluid momentum equations, while the solid mesh is responsible for tracking solid body motion. [60].

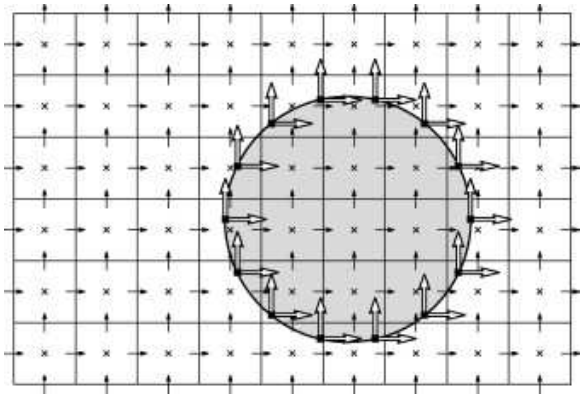

Figure 2.24: Computational domain for the Immersed Boundary Projection method. Image from [61]

Although the original method was developed for elastic FSI problems, several modified implementations have proven its viability to resolve rigid-body FSI problems. The Immersed Boundary Projection Method [61] was successfully applied to model rigid-body FSI problems in which the motion of the immersed rigid structure was prescribed. In this particular study Figure 2.24 shows the discretized physical domain for a cylinder immersed in a viscous fluid. The solid dots over the cylinder surface serve to discretize the rigid body, whereas a staggered variable storage was used for the fluid computational domain. More recently it was successfully applied to model strongly coupled two-way FSI of a set of articulated plates. [62].

One of the earliest published studies employing overset grids to work out FSI phenomena is that of Freitas and Runnels [63]. They analysed the flow-induced rigid-body dynamics of a square cylinder. By integrating the stress field generated by the flow over the surface of the cylinder, the resultant force and moment were fed to the rigid-body dynamics equations to compute the resulting kinematics. In the same vein, Prewitt [64] further extended the Chimera method to account for the six-degrees of freedom rigid-body motion for purposes of aerodynamic analysis.

Another rather interesting combination is the use of unstructured flow solvers along with overlapping grids. Such pairing enhances the capabilities of the already powerful overset method and allows localised grid refinement if necessary. This technique was exemplified in [65] to calculate the flow field subjected to prescribed rigid-body motion interaction. The novelty presented in this research is an alternative approach to treat mesh motion. The Local-time-derivative (LTD) approach disregards any grid motion and solves the flow governing equations in an Eulerian reference frame. This is possible only when employing fully implicit time integration schemes. As a consequence of the implicit nature of time discretization scheme, all fluxes and source terms are computed based on information at the current time step, with only the time-dependent term requiring flow variables from the previous instant in time. The time-dependent term in an Eulerian reference frame describes the rate of change of a property at a fixed point in space, which in this case is the new location of the control volume centre. In recognition of this fact the LTD method is a trade-off between ignoring grid motion in the governing equations, and the need to interpolate the required flow information to the new control volume position to compute the transient term. Another consideration is the need to explicitly include moving wall effects on grid cells. This can be achieved by the addition of the amount of fluid displaced by walls to near moving boundary cells in the pressure correction equation. The LTD method offers considerable advantages over traditional ALE formulations particularly when arbitrary three dimensional polyhedral cells are employed, since the computation of the swept volume by each cell face required to satisfy the space conservation law can become complicated.

The overset-grid technique is increasingly becoming popular within the CFD community to solve rigid-body FSI problems. Recently there has been a surge in the usage of the method to model naval applications. Carrica et al. [66] were amongst the pioneers in this particular field. More recently their research group has implemented a tailored solver in the open source OpenFOAM platform that makes use of overlapping grids to model ship hydrodynamics under different operating conditions [67]. The authors employ Euler angles as parameters to represent rigid-body orientation, whereas the present work employs a more general quaternion approach for such a purpose. Even though the former approach cannot represent all possible body orientations without problems, the naval applications presented in their work are not limited by the gimbal-lock phenomenon arising from the use of Euler angles. Their simulations are impressive in that the problems they tackle require several complex individual components: free surface solver, six degree of freedom solver (6DOF), unstructured overset grids, turbulence modelling. The end result is a model with great level of detail which can even include ship propellers as shown in Figure 2.25.

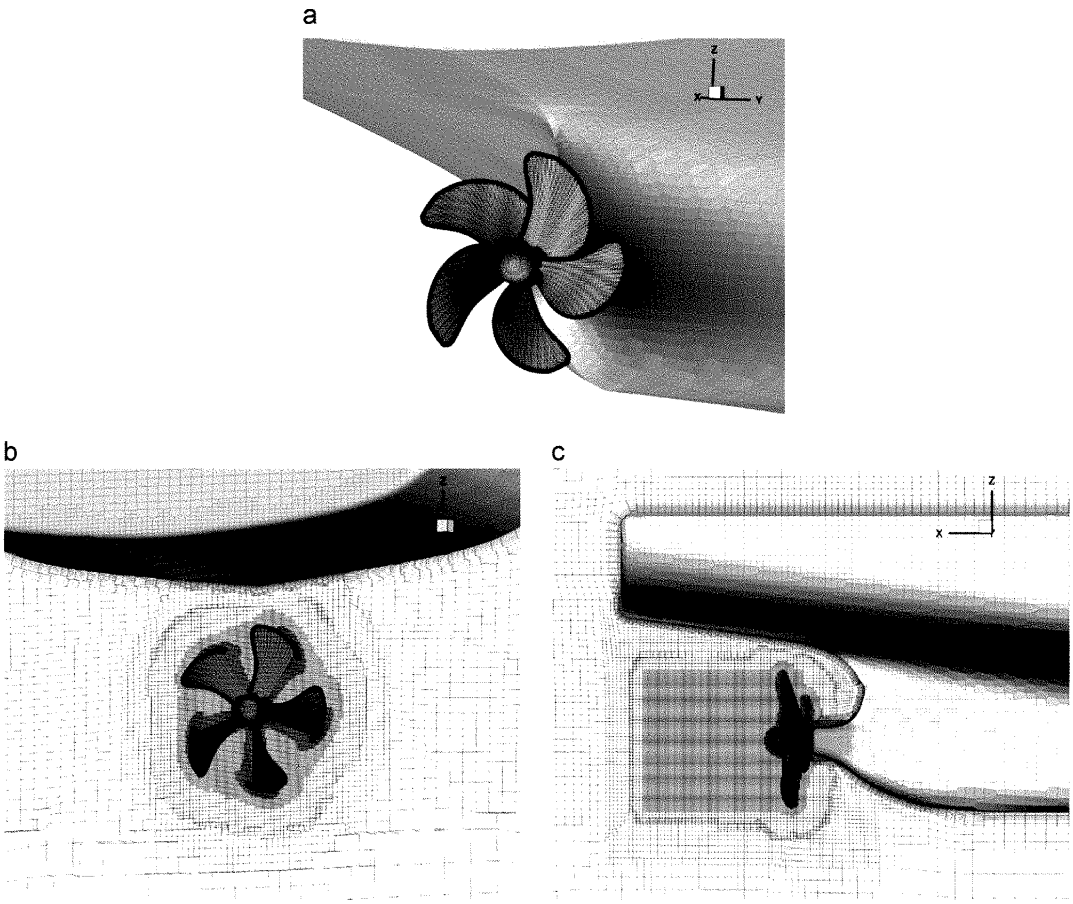

Figure 2.25: Embedded grids for the propeller. Image from [67]

2.6 Conclusions

As demonstrated by the multiple and ever growing number of research projects involving the use of the FVM to solve solid mechanics problems, elastic FSI modelling seems to be pointing towards a partitioned single FVM formulation. The conservative properties of the FVM along with the existence of iterative solvers allow it to tackle fluid dynamics problems more efficiently than its FE counterpart while being perfectly capable of simulating elasticity mechanics. The use of a single integrated environment for simulating solid deformation FSI problems avoids many of the difficulties associated with different data structures and interpolation schemes. Furthermore, readily available open source FVM CFD tools such as OpenFOAM have instilled methodical software development practices suitable for complex undertakings such as FSI simulations.

The use of the overset-grid method provides considerable advantages when it comes to modelling fluid flow around moving bodies. In combination with the ALE method and unstructured grids, it allows to preserve mesh quality even around deforming solid bodies, and enables the flexibility of introducing local mesh adaption techniques. Ultimately, a partitioned single unstructured FVM formulation, coupled with the overset-grid method, proves to be a tough modelling environment to beat for deformable solid FSI problems.

Although not as popular as the Immersed Boundary method, the use of overlapping grids to model fluid flow around moving rigid-bodies is definitely more established than its use to model flow over elastic bodies. All but one of the studies presented in this section regarding moving grids use the ALE approach to discretize the governing equations, nevertheless the LTD approach is an interesting alternative especially for arbitrarily shaped computational cells.

The two main FSI coupling techniques were explored in this chapter. The partitioned approach is found in most applications because it makes use of existing fluid flow solvers, whereas the implementation of monolithic algorithms requires either extensive software modification, or an implementation built from the ground up.

For FSI simulations involving two-way rigid bodies both Runge-Kutta and multistep methods are widely used. Runge-Kutta methods are generally more accurate but might

require additional computational effort. Multistep methods are less computationally intensive. In the particular FSI simulation scenario both techniques are not far apart in terms of computational effort if strong coupling is required.

Chapter 3

Computational Model Development

3.1 Introduction

In this chapter all elements that constitute the computational model of this work are presented. The governing equations pertaining to two-way rigid-body FSI phenomena are introduced in section 3.2. A description of the employed techniques to materialise such a simulation framework is then presented in section 3.3. These include: a) the Finite-Volume method (FVM) and the Hole Cutting algorithm to discretize the fluid governing equations and the physical domain using overlapping grids respectively, b) the Arbitrary Lagrangian Eulerian (ALE) formulation along with the Space Conservation Law (SCL) to properly account for grid motion, c) a rigid-body motion solver, d) and a partitioned predictor-corrector algorithm to couple the interaction between both media. The base steady overset implementation (a) is as described in Skillen [1], and formed the starting point for the present study, which began by extending this for time-dependent flows, and then introduced the elements b) to d) above.

3.2 Governing Equations

3.2.1 Fluid Dynamics

The assumption of a continuous medium is valid when the average distance a molecule of a certain atom travels before colliding with other molecules is of a much smaller order of magnitude than a relevant length scale of the problem being considered. It is then reasonable to assume the distribution of properties in a given medium as continuous functions of position and time. This work makes use of continuum mechanics to model incompressible laminar flow by means of the conservation of mass and momentum.

When working with continuum mechanics the concept of *flux* is rather important. In the realm of transport phenomena the term *flux* is either a tensor of rank 1 (e.g. energy flux) or a tensor of rank 2 (e.g. momentum flux) representing the time rate of a quantity per unit area $\frac{x}{m^2 s}$.The only manner in which something can cross a given surface is if it travels in a perpendicular direction to the surface. Acknowledging the fact that any surface can be represented mathematically by a vector whose magnitude stands for its area, and whose direction is perpendicular to the surface itself, the concept of flux across a surface can be readily envisioned. Fluxes through surfaces play a central role in the formulation of conservation laws for continuum mechanics because they are are mathematical tools to quantify the net balance of the property within a region of interest.

The physical laws governing fluid flow are usually expressed using an Eulerian flow description through the use of the Reynolds transport theorem:

$$\frac{d}{dt}\int_{\Omega_{CM}} \rho\phi(\mathbf{x},t)d\Omega = \frac{d}{dt}\int_{\Omega_{CV}} \rho\phi d\Omega + \oint_{S_{CV}} \rho\phi(\mathbf{U}-\mathbf{U_g})\cdot\mathbf{n}dS \tag{3.1}$$

This theorem is expressed through a differential operator that relates the time rate of change of an intensive property ϕ within a *fixed-mass* moving fluid parcel (left hand side of the equation, Lagrangian flow description), to the time rate of change of the property as seen by an observer focusing on a finite given spatial region (control volume) which might be in motion as well (first term in the right hand side of the equation), plus the net flux of the property across the boundaries of the chosen control volume due to the relative motion between the medium and the control volume (second term in the right hand side

of the equation).

The traditional Eulerian flow description is attained when considering static control volumes in the Reynolds transport theorem, resulting in:

$$\frac{d}{dt}\int_{\Omega_{CM}} \rho\phi(\mathbf{x},t)d\Omega = \frac{\partial}{\partial t}\int_{\Omega_{CV}} \rho\phi d\Omega + \oint_{S_{CV}} (\rho\phi\mathbf{U})\cdot\mathbf{n}dS \tag{3.2}$$

In this case the control volume velocity U_g is zero and the total derivative in the first term of the right hand side of Eqn. (3.1) reduces to a local derivative. This transient term denotes a temporal variation of the property at a fixed point in space, whereas the convective term represents the flux of the property across the boundaries of a static control volume due to bulk fluid motion.

Using an Eulerian flow characterisation by means of Eqn. (3.2), the time rate of change of the intensive property ϕ in control volume C_V is due to *volumetric* and *surface* sources [68]:

$$\frac{\partial}{\partial t}\int_{\Omega_{CV}} \rho\phi d\Omega + \oint_{S_{CV}} (\rho\phi\mathbf{U})\cdot\mathbf{n}dS = \int_{\Omega_{CV}} Q_{\Omega_{CV}}(\phi)d\Omega + \oint_{S_{CV}} \mathbf{Q}_s(\phi)\cdot\mathbf{n}dS \tag{3.3}$$

the most common volumetric sources are due to body forces, chemical reactions, magnetic forces, or heat generation processes. Surface sources stem from molecular driven gradients across the control volume surface of the intensive property.

Eqn. (3.3) is the general conservation equation for any intensive property being transported by a flow field. With this equation in mind, and neglecting any volumetric sources in the linear momentum conservation equation, the physical conservation laws for laminar isothermal incompressible flow are:

- *Conservation of mass*:

$$\oint_{S_{CV}} \rho\mathbf{U}\cdot\mathbf{n}dS = 0 \tag{3.4}$$

- *Conservation of linear momentum*:

$$\frac{\partial}{\partial t}\int_{\Omega_{CV}} \rho\mathbf{U}d\Omega + \oint_{S_{CV}} (\rho\mathbf{U}\mathbf{U})\cdot\mathbf{n}dS = \oint_{S_{CV}} \mathbf{T}\cdot\mathbf{n}dS \tag{3.5}$$

where $\mathbf{T}$ is the molecular momentum flux tensor.

Constitutive relations then complete the system of equations. For Newtonian fluids the ratio of molecular momentum flux to shear rate is constant for a given temperature and pressure [69] and is an important property in fluids named viscosity. Additionally, another source of momentum flux across the surface of the control volume is due to the pressure distribution of the surrounding fluid. These considerations along with the incompressibility assumption allow the molecular momentum flux tensor to be expressed as:

$$\mathbf{T} = -P\mathbf{I} + \mu[\nabla\mathbf{U} + (\nabla\mathbf{U})^T] \tag{3.6}$$

where $\mathbf{I}$ is the unit tensor, and μ is the molecular viscosity.

3.2.2 Rigid-Body Dynamics

In classical mechanics the governing equations of motion are usually formulated by considering a fixed mass region for analysis and tracking its evolution in time due to external interactions. In order to properly quantify these external interactions at least one reference frame has to be defined. Reference frames can be further classified as inertial or non-inertial. An inertial reference frame is subject to zero acceleration [70] which automatically excludes any sort of rotational motion. Furthermore, there is a need to employ two reference frames to describe rigid-body orientation, with one of them being body fixed [70]. If the body exhibits rotational motion this implies the inherent existence of a non-inertial reference frame. Due to this intrinsic availability of two reference frames when dealing with rotating rigid bodies, the analyst has the choice of solving the governing equations in either one of them or to use a combination of both. Since rigid-body motion can be easily segregated in its translational and rotational contributions, a common clever choice is to solve translational body motion in the same inertial reference frame as the fluid governing equations, whilst solving the rotational body motion in the non-inertial body fixed frame [59]. This is done to avoid the computation of a time-dependent moment of inertia tensor which arises when solving body rotation in an inertial reference frame, and it is the approach used in this work as well.

Translational motion of the centre of gravity (COG) is then accounted for by solving

Newton's second law (Eqn (3.7)) in the global inertial reference frame:

$$\mathbf{F} = m\mathbf{a} = m\frac{d\mathbf{V}}{dt} \tag{3.7}$$

For FSI simulations the external force vector includes contributions from the viscous and pressure forces exerted by the fluid on the body, and forces due to mechanical elements such as springs or dampers. As mentioned before, since both fluid and translational solid motion are solved in the inertial reference frame, there is no need to transform the computed fluid forcing vector to solve this ordinary differential equation.

After body acceleration has been computed, translational kinematics (Eqns. (3.8), (3.9)) define the velocity and position of the COG respectively:

$$\mathbf{a} = \frac{d\mathbf{V}}{dt} \tag{3.8}$$

$$\mathbf{V} = \frac{d\mathbf{x}}{dt} \tag{3.9}$$

where $\mathbf{V}$ and $\mathbf{x}$ represent the velocity and position of the rigid body COG.

Rotational kinetics are modelled through Euler's rotational equations (Eqn. (3.10)), which establish a relation amongst body angular acceleration $\boldsymbol{\alpha}$, the body moment of inertia tensor $\mathbf{I}$, body angular velocity $\boldsymbol{\omega}$, and the external torques $\mathbf{T}$ acting on the body which stem from fluid or mechanical interactions.

$$\boldsymbol{\alpha} = \mathbf{I}^{-1} \cdot (\mathbf{T} - \boldsymbol{\omega} \times (\mathbf{I} \cdot \boldsymbol{\omega})) \tag{3.10}$$

$$\boldsymbol{\alpha} = \frac{d\boldsymbol{\omega}}{dt} \tag{3.11}$$

It should be stressed that all terms in Eqn. (3.10) need to be expressed in the body fixed non-inertial reference frame. Once body angular acceleration has been calculated, Eqn. (3.11) is used to update the body angular velocity. Instead of using Euler angles to visualise body orientation as three consecutive rotations about orthogonal axes, it can also be envisioned as a single rotation of θ degrees about a single axis known as the Euler

axis whose unit vector $\hat{e}$ satisfies $\hat{e}_0^2+\hat{e}_1^2+\hat{e}_2^2=1$. Quaternions are then introduced as parameters to populate the direction cosine matrix Eqn. (3.12) which represents the relative attitude between the two reference frames (inertial relative to body-fixed as presented):

$$\begin{pmatrix} 1-2(q_1^2+q_2^2) & 2(q_0q_1+q_2q_3) & 2(q_0q_2-q_1q_3) \\ 2(q_0q_1-q_2q_3) & 1-2(q_0^2+q_2^2) & 2(q_1q_2+q_0q_3) \\ 2(q_0q_2+q_1q_3) & 2(q_1q_2-q_0q_3) & 1-2(q_0^2+q_2^2) \end{pmatrix}^T \tag{3.12}$$

Quaternions are functions of the Euler axis and its rotation angle as shown in Eqn. (3.13).

$$\begin{aligned} q_o &= \hat{e}_0 sin\frac{\theta}{2} \\ q_1 &= \hat{e}_1 sin\frac{\theta}{2} \\ q_2 &= \hat{e}_2 sin\frac{\theta}{2} \\ q_3 &= cos\frac{\theta}{2} \\ 1 &= q_0^2+q_1^2+q_2^2+q_3^2 \end{aligned} \tag{3.13}$$

The use of quaternions has the following advantages over Euler angles: a) all possible body orientations can be truthfully represented avoiding the Gimbal lock phenomenon, b) less computational overhead imposed on the solution algorithm since they do not introduce additional trigonometric functions to the direction cosine matrix, and c) no potential loss of accuracy due to round-off errors as long as they are normalised after being updated. [70]. Rotational kinematic equations are then completed with the calculation of the quaternions by means of Eqn. (3.14).

$$\begin{pmatrix} \dot{q_0} \\ \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{pmatrix}\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} \tag{3.14}$$

3.3 Numerical Solution of the Physical Model

3.3.1 The FVM Discretization Technique

The objective of any discretization technique is to transform one or more partial differential equations into an algebraic or system of algebraic equations [68], [9]. This transformation process inherently introduces approximations, those approximations in turn introduce errors. The error introduced by the discretization practice is known as *discretization error*, and it is equal to the difference between the analytical solution of the partial differential equation and the exact solution of the discretized equation. Any continuum mechanics discretization practice comprises two essential elements: discretization of the physical domain, and discretization of the governing equations [71], [9].

Spatial discretization replaces the continuous nature of the physical domain with a finite set of points which serve as the location where the values of the dependent variables are to be computed. For the particular case of the FVM, each of the points generated by the spatial discretization practice is confined within a volumetric region called *control volume* (CV) or *cell*. For time-dependent problems, time is also discretized. Equation discretization replaces each term of the governing equations with an approximate algebraic expression.

Nowadays almost all commercial fluid flow FVM codes employ collocated grids, implying all dependent variables are stored at the same location within the CV. In early implementations of the FVM flow velocity components were stored at the face centres of the CV (staggered grids). This was done to avoid obtaining non-physical pressure and velocity (for the case of incompressible flow) fields, which would emerge as a result of assuming second order accurate linear profiles for the variation of pressure and velocity within CVs, which would introduce incorrect pressure gradients driving fluid motion. This problem is now traditionally addressed in collocated grids by using a different interpolation practice for the cell face velocities when computing the mass flux through CV faces. The interpolation introduces the effect of the correct driving pressure gradient thus preventing unphysical fields and a widely-used example of such an approach is that proposed by Rhie-Chow [72].

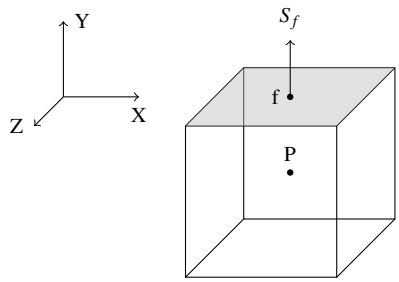

Figure 3.1: Structured control volume topology

The starting point of the current work is an overset steady state implementation of the FVM by Skillen [1]. The base implementation makes use of structured grids in which the solver is capable of handling non-orthogonal hexahedra as control volumes. A sample control volume is shown in Fig. 3.1, the location of the point P where any dependent variable will be stored is at the centroid of the CV such that [68]:

$$\int_{\Omega} (\mathbf{x} - \mathbf{x_P}) d\Omega = 0 \tag{3.15}$$

The face area vector S_f is normal to the face pointing away from the cell centre. The point on the face labelled f is located on its centre, and this location is used to approximate surface integrals using the mid-point rule as shown in Eqn. (3.16).

$$\int_S f dS \approx f S \tag{3.16}$$

Since dependent variables are stored in the cell centre, additional interpolation is required to evaluate their value at the face centroid. Provided that this face interpolation is of at least second-order accuracy, the integral evaluated using the mid-point rule itself is guaranteed to be second order accurate with respect to grid spacing [9].

Considering the generic transport equation for an intensive scalar property ϕ in integral form given by:

$$\underbrace{\frac{\partial}{\partial t}\int_{\Omega_{CV}} \rho\phi d\Omega}_{\text{Transient}} + \underbrace{\oint_{S_{CV}} (\rho\phi\mathbf{U}) \cdot \mathbf{n} dS}_{\text{Advection}} = \underbrace{\int_{\Omega_{CV}} Q_{\Omega_{CV}}(\phi) d\Omega}_{\text{Volumetric sources}} + \underbrace{\oint_{S_{CV}} \mathbf{Q}_s(\phi) \cdot \mathbf{n} dS}_{\text{Surface sources}} \tag{3.17}$$

is a second-order partial integro-differential equation, then the discretized equation should

be of at least second-order accuracy or higher in order to improve the quality of the solution [68]. However, in practice it is not straightforward to formulate a globally second order accurate discretized equation whilst conserving highly desired properties such as *boundedness* and *accuracy*.

The FVM is based on the discretization of Eqn. (3.17) integrated over time as shown in Eqn. (3.18).

$$\int_t^{t+\Delta t}\left[\frac{\partial}{\partial t}\int_{\Omega_{CV}}\rho\phi d\Omega+\oint_{S_{CV}}(\rho\phi\mathbf{U})\cdot\mathbf{n}dS-\int_{\Omega_{CV}}Q_{\Omega_{CV}}(\phi)d\Omega\right]dt = \int_t^{t+\Delta t}\left(\oint_{S_{CV}}\mathbf{Q}_s(\phi)\cdot\mathbf{n}dS\right)dt \quad (3.18)$$

In the following subsections a brief presentation of the discretization of all the terms in Eqn. (3.18) will be made. Due to the implicit nature of the implemented time discretization schemes in this work requiring only previous time values of the dependent variables in the transient term (this will be demonstrated in the time discretization section), the discretization procedure for all spatial terms will be first introduced as though they are independent of time.

Advection Term

The advection term requires the computation of the projection of the intensive property ϕ being transported by the mass flux vector along the surface normal direction for each face bounding the CV. As seen previously the approximation of this term by means of the mid point rule implies obtaining the value of this quantity at the face centre. In this implementation the computation of such a term is done in two stages. In the first stage the face mass flux is calculated by linear interpolation of the cell centre velocity, this velocity is further corrected by means of the Rhie Chow interpolation. The second stage involves the interpolation of the intensive property to the cell face centre. Skillen [1] originally implemented four intra-grid schemes for this purpose but only three have been employed throughout this study: Upwind (zero-degree polynomial), Linear(fist-degree polynomial), and QUICK (second-degree polynomial). Figure 3.2 presents these interpolation polynomials for one dimensional advection when the flow is from left to right.

Upwind interpolation is first order accurate with respect to grid spacing and is the only scheme exhibiting boundedness unconditionally, but it does so at the expense of introducing numerical diffusion. Linear interpolation is formally of second-order accuracy with regard to grid spacing only for uniform grids. However, its error reduction trend as the mesh is refined is quite similar to that of second order accurate schemes for non uniform grids [9]. Quadratic Upwind Interpolation for Convective Kinematics (QUICK) [9] makes use of a parabolic profile to obtain cell face values. In order to do so, information at two consecutive upstream cell centre locations is required. Although it is of third order accuracy, when it is combined with the second order accurate mid point rule to approximate the surface integral, the resulting discretized term is still only second order accurate [9]. The advection term is then discretized as [9] [68]:

$$\oint_{S_{CV}} (\rho\phi\mathbf{U}) \cdot \mathbf{n} dS \approx \sum_f (\rho\mathbf{U} \cdot \mathbf{n}S)_f \phi_f \tag{3.19}$$

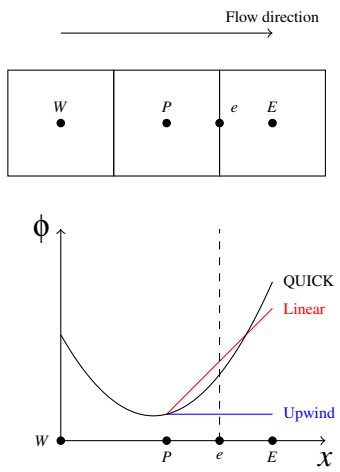

Figure 3.2: Interpolation schemes for the advection term.

Surface Sources Term

Since the surface source term in the general conservation equation (3.17) involves a surface integral, the same discretization procedure using the mid point rule is employed e.g:

$$\oint_{S_{CV}} \mathbf{Q}_s(\phi) \cdot \mathbf{n} dS \approx \sum_f (\mathbf{Q}_s(\phi) \cdot \mathbf{n}S)_f \tag{3.20}$$

However, as seen in Eqn. (3.5) for the case of fluid flow the surface source term is represented by the molecular momentum flux tensor Eqn. (3.6). The introduction of the molecular momentum flux tensor in Eqn. (3.20) results in two distinctive terms, a pressure term and a viscous term.

The pressure term is discretized as:

$$\oint_{S_{CV}} P\mathbf{I}\cdot\mathbf{n}dS \approx \sum_f (P\mathbf{I}\cdot\mathbf{n}S)_f \tag{3.21}$$

with I as the unit tensor and the product $P\mathbf{I}\cdot\mathbf{n}$ representing the momentum flux vector due to pressure distribution through the cell face. The value of the pressure at the face centre is obtained through linear interpolation.

The viscous term is discretized as:

$$\oint_{s_{CV}} \mu(\nabla\mathbf{U}+(\nabla\mathbf{U})^T)\cdot\mathbf{n}dS \approx \sum_f (\mu(\nabla\mathbf{U}+(\nabla\mathbf{U})^T)\cdot\mathbf{n}S)_f \tag{3.22}$$

which requires the evaluation of velocity gradients at the cell face centre. These velocity gradients are evaluated by considering a linear variation of velocity between adjacent control volume centres and approximating the derivative at the face centre using central differences [1].

Time Discretization

The first major contribution of this study to the provided steady state overset solver was to model time-dependent problems. In order to do so, a transient term has to be introduced (see Eqn. (3.17)) and the global conservation equation has to be integrated with respect to time as seen in Eqn. (3.18). Following the discussion presented in [9] regarding time discretization schemes, it is rather insightful to explore the time discretization procedure along the lines of an ordinary differential equation, Eqn. (3.23).

$$\frac{d\phi(t)}{dt} = f(t,\phi(t)) \tag{3.23}$$

Numerical solution to Eqn. (3.23) requires integration in time on both sides for a

given time interval $\Delta t = t_{n+1} - t_n$. The left hand side can be evaluated exactly without introducing any approximation, whereas the computation of the right hand side term can not proceed unless an assumption is made about the evolution of the forcing term $f(t, \phi(t))$ in time as shown in Eqn. (3.24).

$$\int_{t_n}^{t_{n+1}} \frac{d\phi(t)}{dt} dt = \phi^{n+1} - \phi^n = \int_{t_n}^{t_{n+1}} f(t, \phi(t)) dt \tag{3.24}$$

The first time discretization scheme implemented employs a zero-degree polynomial to model the variation of the forcing term with respect to time (i.e. a constant function) and to integrate the right hand side of equation (3.23). Still there are three popular choices one can make belonging to the same family of polynomials: $f(t, \phi(t)) = f(t_n, \phi(t_n))$, $f(t, \phi(t)) = f(t_{n+1/2}, \phi(t_{n+1/2}))$, and $f(t, \phi(t)) = f(t_{n+1}, \phi(t_{n+1}))$. Fig. 3.3 shows graphically the implications of computing the integral(shaded area) using the last two constant profiles in time. The Euler implicit time discretization scheme results from the selection of the last zero-degree polynomial in the previous list. It belongs to the family of implicit methods because integration in time of such profiles results in the evaluation of the dependent variables for all cell neighbour CVs at time t_{n+1} for which the solution is not yet known. Eqn. (3.25) is the result of the time discretization process.

$$\phi^{n+1} - \phi^n = f(t_{n+1}, \phi(t_{n+1}))\Delta t \tag{3.25}$$

This result can be readily extrapolated, and the implicit nature of the scheme still seen, when applied to the general conservation equation. The Euler implicit scheme shares the same discretization properties as its advection counterpart, the upwind interpolation scheme. It is the only one exhibiting boundedness and it is also of first-order accuracy with respect to time discretization.

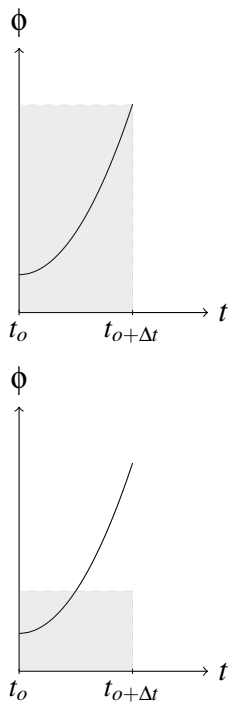

Figure 3.3: Interpolation schemes for the time-dependent term (above: Euler implicit), (below: Mid point rule)

With accuracy in mind, the need to implement a higher order time discretization scheme arose. Amongst the properties sought were improved accuracy over the first order Euler implicit scheme, and a relatively straightforward procedure to implement it. The implicit three-level second-order scheme (also known as Backward differencing in time [68], [9]) fulfills outstandingly such features. It can be derived by approximating *both* sides of Eqn. (3.23) using the mid-point rule centered around time t_{n+1} [9]:

$$\int_{t_{n+1-\frac{\Delta t}{2}}}^{t_{n+1+\frac{\Delta t}{2}}} \frac{d\phi(t)}{dt} dt = \int_{t_{n+1-\frac{\Delta t}{2}}}^{t_{n+1+\frac{\Delta t}{2}}} f(t,\phi(t))dt$$

This then leads to the discretized scheme:

$$\phi_{n+1} = \frac{4\phi_n}{3} - \frac{\phi_{n-1}}{3} + \frac{2f(t_{n+1},\phi_{n+1})}{3}\Delta t \tag{3.27}$$

Again this result is implicit in nature as the forcing term which encompasses the advective, surface sources, and volumetric sources in the general conservation equation is evaluated at the current time t_{n+1}.

Even though this scheme is of second-order accuracy in time, it can be less accurate

than the Crank-Nicholson method depending on the initial error of the selected starter scheme [9] but it is seamlessly implemented in an existing fully implicit framework. As this scheme requires the solution field at two previous time steps an alternative scheme has to be employed at the start of the simulation, when only initial values are known.

The Resulting System of Algebraic Equations

After all terms in the governing equation(s) have been discretized for all computational cells in the grid, a system of algebraic equations can be assembled. A discretized governing equation for a given control volume P can be cast in the following form [9]:

$$A_p \phi_p + \sum_l A_l \phi_l = Q_p \tag{3.28}$$

in which coefficients A_l group the diffusion and advection contributions of the corresponding neighbouring control volumes. The term Q_p incorporates all known values arising from the discretization process which can include the pressure contribution for the case of the momentum conservation equation, and the value of the dependent variable at previous times stemming from the discretization of the time-dependent term. Coefficient A_p accounts for the diffusion, advection, transient, and any other sources depending on the value of the cell currently being considered for computation and should in general satisfy:

$$A_p = A_p^{transient} - \sum_l A_l \tag{3.29}$$

Discretization of the governing equations yields a non-linear algebraic system due to the advection term and possibly to the dependence of the transport properties on the value of the unknown variable (although not the case for flows considered in this project). This system further is linearised by appealing to iterative techniques such as the Picard iteration approach, in which previously computed velocity field values are used to obtain the mass flux required for the advection term, thus effectively allowing the treatment of the advected intensive property ϕ term as linear. In this manner the resulting system of linear algebraic equations can be expressed as Eqn. (3.30) in a matrix-vector system.

$$\mathbf{A}\phi = \mathbf{Q} \tag{3.30}$$

For a structured solver as is the case in this project, the structure of the linearised system of equations **A** is poly-diagonal and sparse [9]. This matrix structure is specially well suited to be tackled by iterative solvers. Skillen [1] implemented the *alternating direction implicit* (ADI) iterative algorithm for the base overset solver in which boundary information is propagated through the domain in three sets of sweeps each along a principal direction for each solver iteration, taking advantage of the structured nature of the computational domain and effectively splitting the problem into three 'one-dimensional' sets. For each set of sweeps a whole row or column of the computational domain is solved at a time along the span of the selected direction. This splitting process further simplifies the structure of the original sparse matrix into a tri-diagonal arrangement which can be solved quite efficiently with the Tridiagonal Matrix Algorithm (TDMA).

Pressure-Velocity Coupling

With the description of the discretization of all the terms in the momentum conservation equation completed, it is now appropriate to introduce a special requirement for the incompressible fluid flow governing equations solution procedure, namely pressure-velocity coupling. Pressure-velocity coupling is required for incompressible flow as the mass conservation equation does not involve the pressure, which does appear in the momentum conservation equation in the form of gradients across cell faces. There is a handful of coupling algorithms which are commonly used in CFD and most of them belong to the family of projection methods. The Semi Implicit Method for Pressure Linked Equations (SIMPLE) [7] algorithm is used for coupling purposes in this project as it was implemented by Skillen [1] for the steady overset solver.

The general approach for this family of coupling algorithms is to build a pressure (or pressure correction) Poisson equation by taking the divergence of the momentum equation [9] and solving it numerically. For this approach to attain its goal of generating pressure and velocity fields abiding by the physical conservation laws, the Poisson equation needs to inherit the discretization characteristics of its constituent elements, i.e. the

continuity and momentum conservation equations.

As noted in the previous section, the present work employs implicit time-discretization schemes. This results in the pressure and viscous terms in the momentum equation being evaluated at the current unknown time step. As such, an initial assumption of the pressure field is required to calculate flow velocity from the linearised momentum equations. It is evident that this computed velocity field will not satisfy the continuity equation and needs to be corrected. This family of coupling algorithms employs corrections of the form:

$$u_i^m = u_i^{m*} + u^{'} \tag{3.31}$$

$$p^m = p^{m-1} + p^{'} \tag{3.32}$$

where superscripts m and $*$ denote the current time step or iteration and the non-mass conserving velocity field respectively.

In order to obtain a relation between the velocity and the pressure corrections, Eqn. (3.31) and Eqn. (3.32) are substituted into the momentum equation yielding Eqn. (3.33)

$$u_i^{'} = \underbrace{-\frac{\sum_l A_{l\,i}^u u_{i,l}^{'}}{A^{u_i}}}_{\text{neighbour corrections}} - \frac{1}{A^{u_i}} \frac{\delta p^{'}}{\delta x_i} \tag{3.33}$$

where δ represents any kind of spatial discretization scheme employed for the pressure term in the momentum equation. The continuity equation is then discretized and combined with Eqns. (3.31) (3.33) to produce the discrete form of the pressure correction equation. This is the general equation from which most of the pressure correction methods stem. Differences amongst these methodologies are related to the treatment of the term labelled as 'neighbour corrections' in Eqn. (3.33). Since this term is not readily available it is neglected in the SIMPLE algorithm and approximated in different ways for coupling procedures such as SIMPLEC, SIMPLER, and PISO. Because the SIMPLE algorithm neglects the so-called neighbour corrections it needs under-relaxation to improve its convergence rate, which was also implemented by Skillen [1]. Eqn. (3.34) is the Poisson equation solved numerically in this work neglecting the neighbour correction

term.

$$\frac{\delta}{\delta x_i}\left[\frac{\rho}{A_p}\left(\frac{\delta p^{'}}{\delta x_i}\right)\right] = \frac{\delta(\rho u_i^{m*})}{\delta x_i} \tag{3.34}$$

The forcing term ρu_i^{m*} represents a spurious mass source for the control volume and it is the divergence producing part that must be projected out of the solution, a summary of the SIMPLE coupling procedure is presented in Algorithm 1

Algorithm 1 SIMPLE algorithm

outer iteration:
Set boundary conditions for the momentum equations
Compute transport coefficients A_i for the momentum conservation equations
Solve the linearised system of algebraic equations to obtain u_i^{m*}
Set pressure-correction boundary conditions and solve pressure-correction equation to obtain $p^{'}$
Correct velocity and pressure fields
goto *outer iteration* until convergence criteria is satisfied

Boundary Conditions

Two types of numerical boundary conditions have been implemented in the original steady state overset code. These are Dirichlet conditions which directly determine the value of the intensive property at the boundary (Eqn. (3.35))

$$\phi_{boundary} = \phi_{Dirichlet} \tag{3.35}$$

, and Neumann conditions which set the gradient of the intensive property in the boundary normal direction equal to zero (Eqn. (3.36)).

$$\left(\frac{\partial \phi}{\partial n}\right)_{boundary} = 0 \tag{3.36}$$

Different sets modelling common conditions encountered by fluid flow at boundaries can be created by combining these two numerical boundary conditions. Inlet boundary conditions impose a given velocity profile as Dirichlet values on a selected boundary for the momentum equations. Wall boundary conditions also set Dirichlet values on a given surface for the momentum equations which reflect the kinematic conditions imposed by the wall on the fluid. Symmetry and outlet boundary conditions establish a zero gradient

normal to the surface for the momentum equations, with the former also imposing zero boundary normal velocity.

At all domain boundaries pressure is required in order to compute part of the molecular momentum flux through control volume faces adjacent to the boundary. This boundary pressure value is extrapolated from the interior of the domain. Boundary conditions for the pressure-correction equation are also required. If pressure is prescribed at a domain boundary then the pressure correction at that location is zero. On the other hand, if a mass flux is prescribed at a boundary then it does not need to be corrected in the pressure-correction equation which results in a zero gradient normal to the boundary for the pressure correction.

3.3.2 Aspects of an Overset Implementation

This section is a brief summary of the overset implementation by Skillen [1] to gain some insight into the essential elements required to solve CFD problems using overlapping grids.

The Hole Cutting Algorithm

The purpose of the hole cutting algorithm is to identify cells in each sub-grid in order to properly establish the required computational stencil to compute the dependent variable values for each control volume. Within the overlap region of an overset grid there can be three different types of control volumes:

- *Standard Cells*
- *Interpolation Cells*
- *Unused Cells*

Standard cells are control volumes whose dependent variables can be calculated by using neighbouring information from either other standard cells or from interpolation cells. *Interpolation* cells are control volumes for which the dependent variable values are obtained not from solving the discretized governing equations, but from interpolating the

values from an interpolation stencil. They also can be required to complete the computational stencil of another standard cell. *Unused* cells are control volumes that do not participate in the computation of the flow field at the current time step —either because they lie outside the flow domain, or because they overlap cells from other sub-grids that are better suited to perform roles as standard cells or interpolation cells at the current time step.

Consider Fig. 3.4, this set of overlapping grids is generated to model flow around a two dimensional circular cylinder with the red grid intending to take precedence over the green background mesh in the region close to the cylinder. The first step for cutting a hole in an overset grid is to determine whether a cell lies partially or totally outside the domain of interest at the time. This can be achieved by testing for the intersection of any edge of a given cell with the edge of the physical domain. To check this, the face which fully lies on the surface of the physical boundary and belongs to the cell adjacent to the boundary within the computational domain provides for a benchmark surface to test for intersections with other hexahedral cells. For illustration purposes Figure 3.5 highlights the physical domain bounding faces in blue for a case where there is an intermediate grid extending over the wall boundaries. In this case all cells belonging to the intermediate grid will be tested for intersection with the blue marked cells.

In recognition of the fact that any quadrilateral face can be split in two triangles, the aforementioned intersection test is at its root a triangle-triangle intersection test. Considering that a control volume has six faces, the test has to be completed a total of twenty-four times for each tested cell (two triangles per boundary face need to be tested for intersecion with two triangles for each of the six faces of the control volume). If an intersection occurs, then the cell is identified as “unused”. Figure 3.6 shows an overset mesh with the cells that intersect boundaries of the computational domain removed via the triangle-triangle intersection test.

As seen in Figure 3.6, there are still cells belonging to the background mesh that are totally out of the computational domain. The next step then is to remove the remaining background grid cells fully lying out of the physical domain. In order to do so, for each cell adjacent to the previously tagged unused cells the algorithm tries to find a suitable

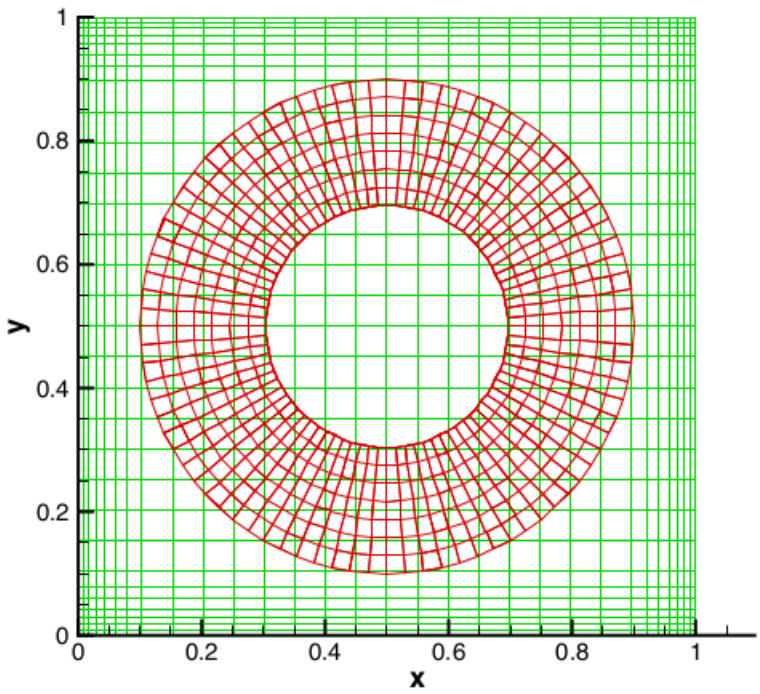

Figure 3.4: Initial overlapping grids.Image from [1]

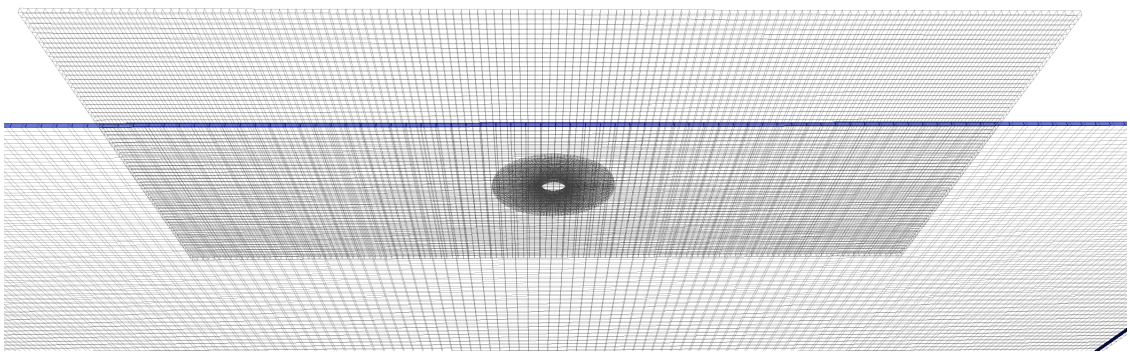

Figure 3.5: Reference cells used for the triangle-triangle intersection test.

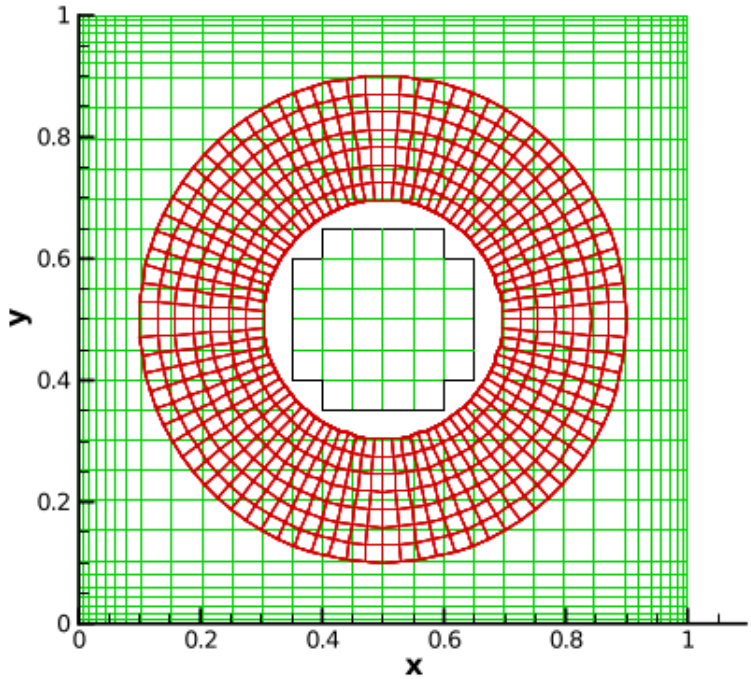

Figure 3.6: Overset mesh after the cell intersection test.Image from [1]

donor mesh from where the dependent flow variables can be interpolated. If no donor mesh can be found then the cell in question must be an unused cell. Otherwise it is labelled as an interpolation cell, Figure 3.7 shows the outcome of this step with the interpolation cells for each grid highlighted.

Although the overset mesh generated in the previous step is good for computational purposes, it is not optimal with regard to computational efficiency since in the overlap region more resources are needed in order for the grids to exchange information. With that in mind, it is possible to reduce the overlap area by establishing a certain criteria to choose which cells to preserve in a given zone. As pointed out by Skillen [1], most of the available hole cutting algorithms employ a relatively simple criterion for this purpose; they assign sub-grid priorities and that way the algorithm decides to exclude some of the cells belonging to the low priority grid in the overlap region. However, this rather simple approach is not always suitable and can lead to a degradation of the overall quality of the mesh by removing cells on low priority grids which may have been especially built for a specific function such as near wall cells.

One of the novelties of Skillen's work was to formulate an algorithm in which it is

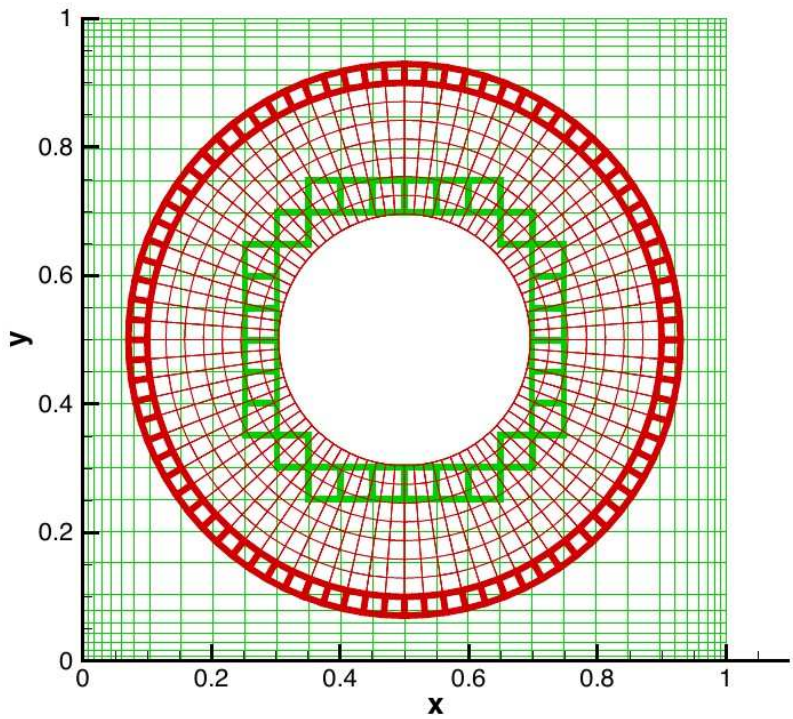

Figure 3.7: Valid non-optimal overset mesh. Interpolation cells highlighted in both grids. Image from [1]

possible to establish a tailored user-defined criteria on a cell by cell basis for the reduction of the overlap zone, for example trying to preserve cells with the lowest possible volume. Although this approach ultimately will lead to a nicely optimised overset mesh, its implementation involves several considerations before the cell removal process can proceed. All cells belonging to a sub-grid that fully overlap in a given zone must not be removed if they are part of any interpolation stencil for a given interpolation cell on another sub-grid. In addition, any cell adjacent to a cell identified previously as required by an interpolating cell in another sub-grid can not be removed either. Furthermore, if a cell is to be removed, any surrounding standard cell would need to be converted to an interpolation cell. If it is not possible to find a donor sub-grid for the standard cell to be converted, then the cell can not be removed. Finally, if the cell to be removed has passed all the mentioned considerations so far, it will be removed if and only if it does not satisfy the user specified criteria when comparing it to the average value of the same criteria of all the surrounding cells in the grid it overlaps. Figure 3.8 shows the resulting overset mesh with the interpolation cells of each grid highlighted.

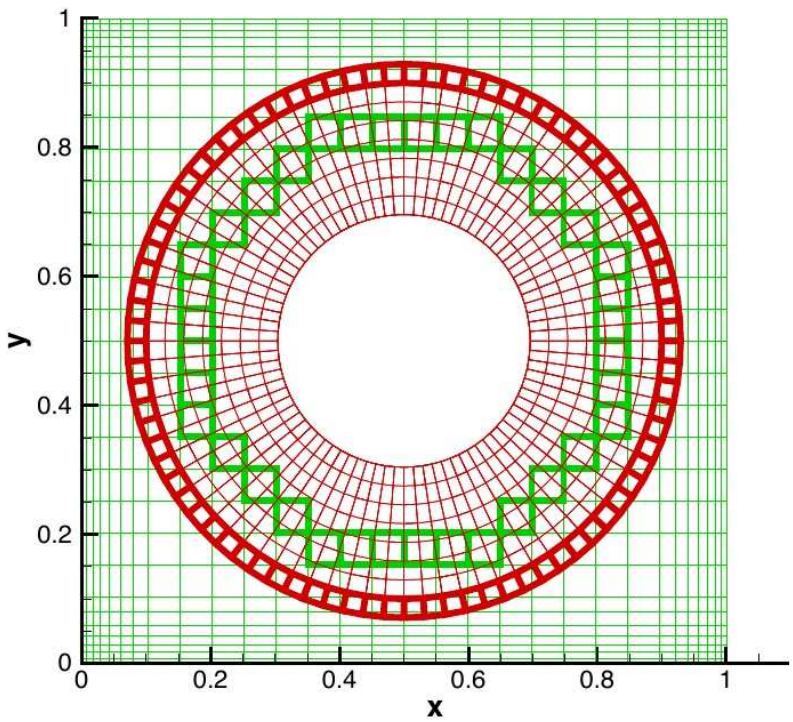

Figure 3.8: Final overset mesh.Interpolation cells highlighted in both grids. Image from [1]

Inter-Grid Communication

The inter-grid communication process amongst overlapping grids is responsible for providing an 'interface' over which information from a donor mesh is transferred to a receiving grid, so that the flow field can be computed in the latter, and it is of crucial importance as the quality of the solution heavily depends on it. This section presents a concise recapitulation of the techniques implemented for this purpose by Skillen, a more detailed discussion on this topic can be found in the literature review section of this thesis 2.4 and of course in [1].

Bilinear or trilinear interpolation can be selected by the user for two-dimensional or three dimensional flow field calculations respectively. Interpolation stencils for linear interpolation comprise eight donor cells whose centres define the vertices of a hexahedron. For each donor grid the number of possible interpolation stencils is $(N_i+1)x(N_j+1)x(N_k+1)$, where N_i represents the number of cells of the donor grid in the i direction. Skillen convincingly makes a case against a straightforward but rather costly algorithm to determine the appropriate interpolation stencil, which would involve visiting each potential stencil at a time, computing the volume of twelve tetrahedra (two per hexahedron

face) formed with a triangular base made up of three vertices of each stencil face and the desired interpolation location $P(x,y,z)$, adding them up, and comparing the result with the volume computed from the position of the vertices of the interpolation stencil. If both calculated volumes are equal it means the interpolation stencil encompasses the desired interpolation location $P(x,y,z)$. Once the stencil is identified the interpolation coefficients would then need to be calculated.

In addition of the computational overhead the above algorithm would imply, it does not guarantee correct results for cases when the interpolation point P would lie close to the faces of the computational stencil. This is due to the fact that the faces of the hexahedron are assumed to be planar and the interpolation point could actually lie between the actual curvature of the hexahedron face and the assumed planar surface.

Instead, the selected algorithm relies on binary search trees as a more efficient way of narrowing down all possible interpolation stencils in a donor grid (see [1] for details). Binary search trees are special data structures which are constructed with the aim of reducing the time it takes to find a particular data element belonging to a data set [73]. A binary search tree structure is constructed by storing data in a non-sequential manner in the computer's memory. As seen in Fig. 3.9 data is then connected by means of pointers, each data item can have two of these pointers which lead to another data item forming a node. In this way, every node except for the first one will eventually have a pointer leading to it. Starting with the first node, every level of connectivity is assigned a number, thus creating a hierarchical data structure. Fig. 3.9 shows a schematic of the possible memory storage array, and a binary search tree with four hierarchy levels. The hierarchy itself is a key element for finding a data item within the structure. Binary search trees form the basis of the Alternating Digital Tree algorithm (ADT) [74] which in turn is the underlying algorithm of the novel geometric intersection ADT proposed by Skillen to determine potential interpolation stencils. The ADT is a special case of the more general binary search tree where the difference lies in the search criteria used to go through the hierarchical data structure.

This procedure yields the actual interpolation stencil for Cartesian grids and a low number of candidate stencils for non-Cartesian grids. For the general case of having more

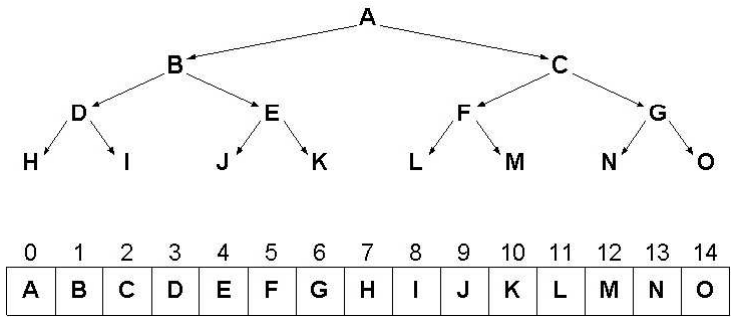

Figure 3.9: Binary search tree and a possible arrangement of nodes in computer memory. Image from [1]

than one potential stencil the interpolation coefficients are calculated as each stencil is evaluated. If the computed coefficients are all in the interval $[0,1]$ the correct interpolation stencil is identified.

For the current dependent variable ϕ, at point x, y, z the interpolation leads to an expression:

$$\phi \approx \sum_{i=1}^{8} N_i \overline{\phi}_i \tag{3.37}$$

and the factors N are obtained by solving:

$$x = \sum_{i=1}^{8} N_i \overline{x}_i \tag{3.38}$$

$$y = \sum_{i=1}^{8} N_i \overline{y}_i \tag{3.39}$$

$$z = \sum_{i=1}^{8} N_i \overline{z}_i \tag{3.40}$$

with the Newton-Raphson method. The coordinates $\overline{x}_i \overline{y}_i \overline{z}_i$ being the locations of the vertices that constitute the interpolation stencil.

The non-conservative nature of linear interpolation is due to the location where it takes place rather than the use of a linear shape function. An alternative methodology to linear interpolation for inter-grid communication was introduced in 2.4, and was implemented by Skillen to strive towards a conservative formulation. The Mass Flux Based Interpolation (MFBI) is a second order approximation with respect to grid spacing of the mass conservation constraint at grid interfaces. This method aims to provide interpolated values of the flow velocity field at cell centres in such a way that the averaged mass flux through the interpolation boundary is in line with mass conservation. Its use improves the

overall convergence rate of the solution when compared to linear interpolation. Details of the actual implementation of the methodology are out of the scope of this work and can be found in [1].

Another aspect that deserves attention when working with moving overlapping grids is the situation described in 2.4 regarding the emergence or vanishing of cells in the computational domain due to the movement of body fitted grids. In this work, an implemented modification of the linear interpolation algorithm searches for adequate interpolation stencils comprising cell centre values available from previous time steps. This applies for cells that were unused at a previous time step and become part of the active domain for the computation of the flow field at the current solution time. The problem of the emergence or vanishing of cells can be largely avoided with grids of comparable sizes in the overlapping zone and with adequate selection of the time step. Since both factors are required anyway to obtain good quality solutions, this issue has been seldomly encountered throughout the set of problems tackled in this thesis.

A SIMPLE Overset Algorithm

A simplified general workflow of the implementation by Skillen [1] is presented here to gain insight into the numerical solution process of the governing equations using overlapping grids.

- Set all boundary conditions, which include interpolation boundary values for each grid in the computational domain.
- Compute the non-mass conserving velocity field u_i^* in each sub-grid for all active cells at a time by solving the discretized momentum equations.
- Calculate the uncorrected mass fluxes through all active or interpolation cell faces using Rhie-Chow interpolation.
- Obtain the pressure correction throughout the computational domain by solving the pressure-correction equation for all active cells in each sub-grid. For interpolation cells the value is interpolated.
- Correct the velocity and pressure fields.

The core of the algorithm is the same as that for non-overlapping grids. The main differences for overlapping grids reside in the need to solve sequentially the fluid flow problem in each sub-grid, and the need to update required boundary information to transfer information to the next grid.

3.3.3 Arbitrary Lagrangian-Eulerian Method

Fluid-solid interaction phenomena are characterised by the coupled behaviour of two (or more) continuous media. The main effect of this intrinsic relationship on the fluid medium is a constant geometrical change of its domain. In order to be able to predict the flow field under these circumstances, the computational domain needs to adapt to these changes as well by moving. Hence, there is a need to formulate the governing equations accounting for such grid movement. This section presents the theoretical framework of the selected formulation implemented in this project to cast the fluid flow governing equations in an *inertial reference frame* for moving control volumes.

In general, as seen in the literature review section of this work, there are two ways of tackling a problem involving moving grids. For this research the Arbitrary Lagrangian Eulerian [41] method is used to account for grid motion in the formulation of the conservation laws. It is important to stress that the fluid flow governing equations are solved in an inertial reference frame, otherwise Coriolis forces would need to be considered for cases involving rotation. With this idea in mind, the Eulerian flow description conservation equation for a given scalar (see Eqn. (3.3)) sees two modifications. The first one pertains to the time-dependent term, whose duty now is to quantify the rate of change of the amount of a property contained in a *moving* (notice the use of the total derivative) control volume due to surface interactions, volumetric source terms , and possibly due to the fact that the control volume itself may change its size with time. The second one is the introduction of the relative convective flux, which is the flux an observer moving with the control volume would see through its boundaries due to advection. With these considerations the integral form of the conservation equation for a given scalar is now:

$$\frac{d}{dt}\int_{\Omega_{CV}} \rho\phi d\Omega + \oint_{S_{CV}} \rho\phi(\mathbf{U}-\mathbf{U_g})\cdot\mathbf{n}dS = \int_{\Omega_{CV}} Q_{\Omega}(\phi)d\Omega + \oint_{S_{CV}} \mathbf{Q}_s(\phi)\cdot\mathbf{n}dS \tag{3.41}$$

In Eqn. (3.41) the computation of the relative convective flux introduces the velocity with which the grid moves. One may intuit the procedure to calculate the relative convective flux is straight forward if the location of the grid is known beforehand at all times. The procedure would involve obtaining the grid velocity at the cell face centre based on its known position vector during the time interval:

$$\mathbf{U_g} = \frac{\mathbf{r}^{n+1}-\mathbf{r}^{n}}{\Delta t} \tag{3.42}$$

if the grid does not deform then Eqn. (3.42) is appropriate and the ALE formulation can be considered complete . However, as it was shown in [75], for the general case of grid deformation artificial mass sources will be introduced in the flow field if this approach of calculating the grid velocity is used. It must be mentioned that there are more complex methods of calculating the grid velocity which do not introduce artificial mass sources. Some of those alternative approaches are reviewed in[76].

The name of the ALE methodology reflects the behaviour of the flow description adopted according to Eqn. (3.41). If control volume velocity is zero then an Eulerian flow characterisation is employed, if control volume velocity is equal to the flow velocity a Lagrangian flow description is used, and if the control volume is in arbitrary motion a hybrid flow description is employed. It is important to note that the flow description method employed is independent of the choice of reference frame to solve the governing equations. For instance, one can adopt a Lagrangian flow description and formulate the governing equations in either an inertial or non-inertial reference frame. It is a good exercise to reflect on this as one tends to automatically associate the term ‘moving grids’ with solving the governing equations in a reference frame moving with the mesh, which could be non-inertial if the mesh is subject to rotational motion. If that is the case the governing equations need to include ’fictitious’ forces arising from the use of the non-inertial reference frame. As previously stated in this section, in this work the fluid flow

governing equations are formulated and solved in a inertial reference employing the ALE a methodology to account for grid motion.

The Space Conservation Law

In order to avoid introducing artificial mass sources to the flow domain one must ensure the Space Conservation Law (SCL) (also referred to as Geometric Conservation Law GCL [48], [9]) is satisfied. The SCL describes the conservation of space when the control volume changes its shape and/or position with time. It can be thought of as the continuity equation in the limit of zero fluid velocity [9]. Its integral form is:

$$\frac{d}{dt}\int_{\Omega_{CV}} d\Omega - \oint_{S_{CV}} \mathbf{U_g}\cdot\mathbf{n}dS = 0 \tag{3.43}$$

The SCL can be enforced by solving numerically the corresponding equation together with the other conservation equations as was done in [77]. The alternative approach is to calculate the grid velocities in a manner consistent with the other conservation equations to ensure the SCL is always satisfied [75]. The latter approach is now commonly used and it is the way it shall be done in this work.

At this point it is worth inspecting Eqn. (3.43) and interpreting its physical meaning. It relates the volume time rate of change of a moving computational cell, to the net volumetric time rate through its boundaries due to its motion. That is, the grid velocity can be interpreted as a flux of volume whose units are $\frac{m^3}{m^2 s}$, and when dotted with the surface area vectors the net amount of *volume* per unit time crossing the cell surface is obtained. It is then suitable to compute the second term as:

$$\oint_{S_{CV}} \mathbf{U_g}\cdot\mathbf{n}dS = \frac{\sum_c \delta\Omega_c}{\Delta t} \tag{3.44}$$

with $\delta\Omega_c$ representing the volume swept by face c during the time interval Δt (see Fig. 3.10). In this way the relative mass flow rate term in Eqn. (3.41) is split and computed as the difference between the absolute mass flow rate, and the mass flow rate due to mesh motion, evaluated using the volume swept by the cell face instead of the grid velocity as shown in Eqn. (3.44). The semi - discretized continuity equation for moving control volumes using

a first order implicit Euler scheme is then:

$$\frac{\rho\Omega^{n+1} - \rho\Omega^{n}}{\Delta t} + \sum_{c} \dot{m}_c = 0 \tag{3.45}$$

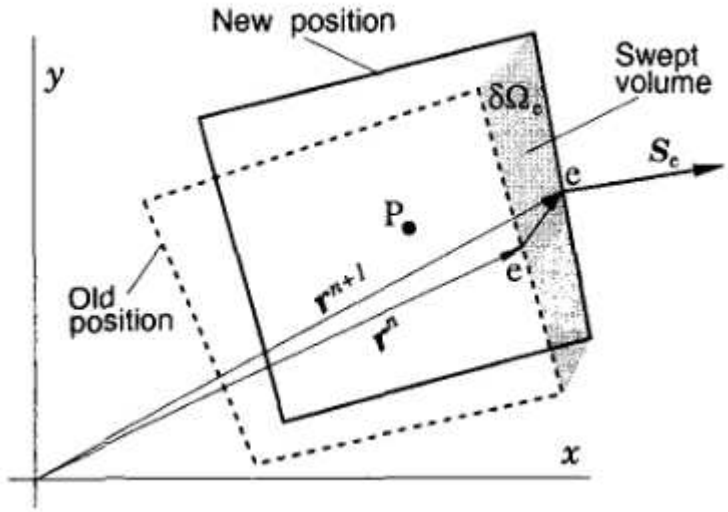

Figure 3.10: Volume swept by a control volume face for a two-dimensional control volume. Image from [9].

It is clear that the contribution of the volume swept by all cell faces shall cancel the change of volume by the cell experienced in one time step, therefore satisfying the SCL.

Another aspect requiring consideration at this point is the implementation of moving wall boundary conditions. For cells adjacent to moving bodies, the no slip condition requires the fluid velocity to be the same as the moving body velocity at the interface between the two media. This means the *fluid* velocity at the cell face which is in contact with the moving body will be different from zero. If the body movement direction is normal to its surface, the mass flow rate through the adjacent cell face would be non zero. However, in such cases the governing equations are solved considering grid motion and the mass flow rate computed at cell faces is the *relative* mass flow rate. Under this consideration the cell face velocity will also be equal to the moving body velocity and thus the relative mass flow rate through the face will be zero. This satisfies both the no slip condition and the zero mass flow rate through impermeable walls.

3.3.4 Implementation of the Space Conservation Law

As seen in the previous section, in this work the implementation of the SCL boils down to calculating the volumes swept by all the faces of the control volumes during a given time interval. In this research the method of calculating the volume swept by a cell face is the one used in OpenFOAM® where it was originally implemented in [78].

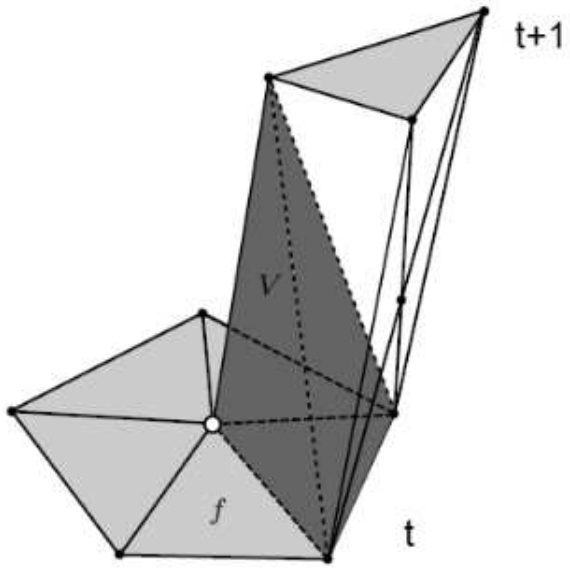

Figure 3.11: Swept Volume of a decomposed face. Image from [78]

The first step is to decompose the face into triangles using its centroid. The volume swept by a triangular face forms a prism with a triangular base. The next step is to decompose the prismatic volume into a set of three tetrahedrons. The first tetrahedron is then formed by connecting any one of the vertices at the initial time with two opposite vertices at the final time as shown in Fig. 3.11. The remaining space in the prism will be made up of a quadrilateral plane and the other two tetrahedrons, which can be formed by connecting one of the diagonals of the plane. The volumes of the three tetrahedrons are then summed and they represent the volume swept by a triangle of a given face.

As noted in [76] the decomposition just described can take place by connecting one diagonal at a time in two unique ways (see Fig. 3.12). Therefore the swept volume is calculated in both ways and the mean value is used as the volume swept by the triangular face.

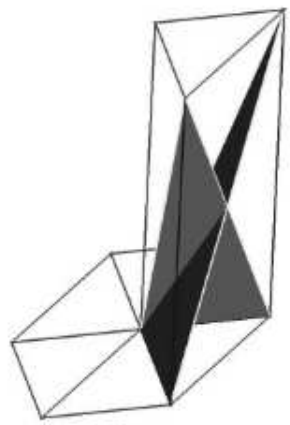

Figure 3.12: Two possible ways of evaluating the swept volume . Image from [76]

3.3.5 Solving the Rigid-Body Dynamics Problem

The structured programming methodology employed in this project to solve the rigid-body dynamics problem is an adaptation of the conceptualisation presented in [79]. The main idea is to update the *state* of a rigid body as it is subject to forcing conditions. In this sense, the state of a rigid body involves defining its position, orientation, linear, and angular momentum at every instant in time. For every rigid body in the simulation its associated body fitted grid(s) is/are evolved according to the resulting kinematic conditions.

The equations presented in section 3.2.2 are a set of 13 ordinary differential equations , and are solved using a predictor-corrector approach which naturally suits the FSI coupling algorithm (presented in the next section). Two numerical schemes belonging to this family have been implemented; Heun's method (second order accurate), and a two-time level method (third order accurate).

Heun's method comprises an Euler-explicit predictor (Eqn. (3.46)) and a trapezoid-rule corrector (Eqn. (3.47)):

$$y_{pred} = y_o + \Delta t f(y_o, t_o) \tag{3.46}$$

$$y_{corr} = y_o + \frac{1}{2}\Delta t \left[f(y_o, t_o) + f(y_{pred}, t_f) \right] \tag{3.47}$$

The two-time level comprises an Adams-Bashforth explicit predictor (Eqn. (3.48)) with an Adams-Moulton corrector (Eqn. (3.49)):

$$y_{pred} = y_o + \frac{1}{2}\Delta t\,[3f(y_o,t_o) - f(y_{oo},t_{oo})] \tag{3.48}$$

$$y_{corr} = y_o + \frac{\Delta t}{12}\left[5f(y_{pred},t_f) + 8f(y_o,t_o) - f(y_{oo},t_{oo})\right] \tag{3.49}$$

3.3.6 The Overset FSI Coupling Algorithm

This section details the last component of a two-way overset FSI implementation, namely the partitioned FSI coupling algorithm. The aim of the algorithm is to achieve a strongly coupled solution of a two-way FSI problem at every time step throughout the duration of the simulation. Although strongly coupled solutions are always desired, the iterative nature of the algorithm allows to easily modify the degree of coupling by either setting the number of outer iterations explicitly, or by specifying less strict convergence criteria.

Fig. 3.13 shows a flowchart of the implemented algorithm. Starting with an established flow field the forces and torques exerted by the flow over the rigid body provide the necessary forcing term to solve the rigid-body governing equations. These forces are computed by integrating the stress tensor $\mathbf{T}$ over each boundary cell face. The resulting force is then the sum of these integrals over all boundary cells, and can be split into pressure and viscous contributions as follows:

$$\mathbf{F}_{pressure} = \sum_{bf}\left(-\int_S p\mathbf{I}\cdot\mathbf{n}dS\right) \tag{3.50}$$

$$\mathbf{F}_{viscous} = \sum_{bf}\left(-\int_S \mu\left(\nabla\mathbf{U} + (\nabla\mathbf{U})^T\right)\cdot\mathbf{n}dS\right) \tag{3.51}$$

The negative sign preceding Eqns. (3.50) and (3.51) reflects the fact that these forces are exerted by the fluid *on* the body. Integrals are discretized using the standard mid-point rule.

Similarly, the required torques acting on the body are calculated as:

$$\mathbf{T}_{pressure} = \sum_{bf}\left(\mathbf{r}_f \times -\left(\int_S p\mathbf{I}\cdot\mathbf{n}dS\right)\right) \tag{3.52}$$

$$\mathbf{T}_{viscous} = \sum_{bf} \left(\mathbf{r}_f \times - \left(\int_S \mu \left(\nabla \mathbf{U} + (\nabla \mathbf{U})^T \right) \cdot \mathbf{n} dS \right) \right) \tag{3.53}$$

where r_f is the position vector from the centre of gravity of the rigid body to the boundary face centre. It is important to note that the torques computed with Eqns. (3.52), (3.53) are in the inertial reference frame. Since the governing equations for rotational motion are solved in the body reference frame, these torques further need to be expressed accordingly using the rotation matrix:

$$\mathbf{T}^b = R^{b/I} \mathbf{T}^I \tag{3.54}$$

with superscripts b and I denoting the body attached , and the global inertial, reference frames respectively.

After solving the predictor rigid-body motion equations, the moving grids are relocated to their new position. In order to do so, the new position vector of a given vertex p belonging to the moving grid is calculated in the inertial frame I as :

$$\mathbf{r}^{p/I} = \mathbf{x}_{COG} + R^{I/b} \mathbf{r}^{p/b} \tag{3.55}$$

where $R^{I/b}$ is the transpose of the rotation matrix introduced in Eqn. (3.54), and $\mathbf{r}^{p/b}$ is the position vector of vertex p expressed in the body attached reference frame b. The hole cutter algorithm is then invoked to completely define the computational domain at the new grid location. Boundary conditions for the fluid flow are then set using the calculated rigid-body boundary kinematic field.

The flow field at the predicted rigid-body location is now computed, and the forces exerted by the flow on the rigid body are determined again. These forces are now used to correct the predicted state of the rigid body. At this point a decision whether to correct the position again, or to proceed to the next time step, needs to be made. Generally speaking, this decision can be based on a normalised quantity reflecting changes in the state of a rigid body between consecutive predictor-corrector or corrector-corrector steps. Alternatively a fixed number of outer iterations can be set as a convergence criterion. If a weakly coupled algorithm is required then only the predictor step is required to advance the simulation time. Regardless of the outcome of the convergence check the flow field needs to be solved again using the last rigid-body corrector kinematic information (otherwise

there would be no point in performing the last correction), this entails the redefinition of the computational domain by moving the grids to the corrected location, and defining the computational domain yet again via the hole cutting algorithm. The bulk of all the actions just described constitute an *outer iteration* for the coupling algorithm. This coupling sequence of actions resembles the predictor-corrector nature of the SIMPLE algorithm to solve fluid flow, and is sometimes referred to as a SIMPLE coupling algorithm [59].

Some FSI problems are challenging to solve because the strong interdependence between solid and fluid domains can lead to violent changes of the coupling conditions at the interface during the predictor-corrector or corrector-corrector stages within outer iterations. For rigid-body FSI problems this leads to non-physical oscillations of the solid body kinematic state, ultimately preventing convergence of the coupling algorithm. This is quite common in deformable solid FSI problems and relaxation techniques have been devised to obtain physical solutions. In this work fixed and adaptive interface relaxation has been implemented not only to prevent convergence issues but also to improve the convergence rate of the algorithm.

Both fixed and adaptive methodologies relax body acceleration in this study. In fixed relaxation the user selects a relaxation factor α, with the resulting acceleration computed as:

$$\mathbf{a}^l = \mathbf{a}^{l-1} + \alpha\left(\mathbf{a}^l - \mathbf{a}^{l-1}\right) \tag{3.56}$$

Adaptive relaxation is implemented as presented in [80] (see Eqns. (3.57)-(3.61)) also known as Aitken relaxation:

$$\mathbf{a}^l = \mathbf{a}^{l-1} + \alpha^l\left(\mathbf{a}^l - \mathbf{a}^{l-1}\right) \tag{3.57}$$

$$\alpha^l = 1 - \gamma^l \tag{3.58}$$

$$\gamma^l = \gamma^{l-1} + \left(\gamma^{l-1} - 1\right)\frac{\left(\Delta\mathbf{a}^{l-1} - \Delta\mathbf{a}^l\right)\cdot\Delta\mathbf{a}^l}{\left(\Delta\mathbf{a}^{l-1} - \Delta\mathbf{a}^l\right)^2} \tag{3.59}$$

$$\Delta\mathbf{a}^l = \mathbf{a}^{l-1} - \mathbf{a}^l \tag{3.60}$$

$$\Delta\mathbf{a}^{l-1} = \mathbf{a}^{l-2} - \mathbf{a}^{l-1} \tag{3.61}$$

this method is based on the use of two previous rigid-body kinetic states to improve the

current solution [81]. As such, for the first corrector iteration of every time step the method employs a fixed relaxation factor selected by the user at run-time. The methods presented here provide six relaxation factors accounting for three linear accelerations and three angular accelerations.

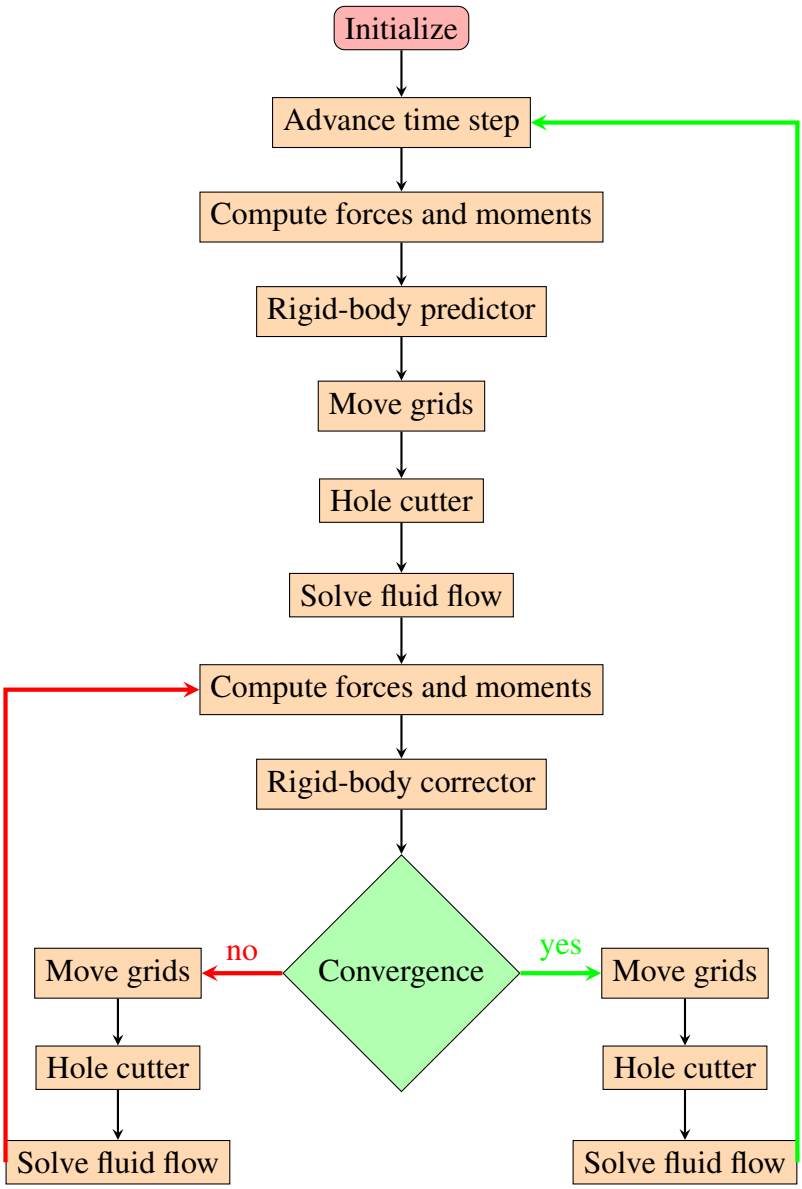

Figure 3.13: Partitioned Coupling Algorithm

3.4 Closure

The governing equations to model rigid-body FSI phenomena have been introduced in this Chapter. Additionally, a concise review of the key numerical techniques required for the current implementation has been presented. The material in this section has gone into sufficient detail to clearly identify the two limiting factors in terms of computational throughput for a two-way FSI application: the iterative nature of the partitioned coupling algorithm to achieve strongly coupled solutions, and serial execution of the fluid flow solver.

Chapter 4

Code Verification

4.1 Introduction

Code verification is the main subject of this Chapter. The essential components of the overset FSI implementation have been tested individually. This process demonstrates the ability of the numerical method to produce mathematically correct solutions, and provides an estimate of the order of accuracy of the application under different conditions.

The principal methodology to perform code verification is introduced in section 4.2. As a starting point, non-overlapping computational domains are employed to establish a reference in terms of order of accuracy in section 4.3. The effect of inter-grid communication in the order of accuracy of the overset method is then presented in section 4.4, and verification of the Space Conservation Law is shown in section 4.5. Finally, the results of the rigid-body solver verification process are examined in section 4.6.

It is worth mentioning that thorough code verification implies testing for *all* possible setting combinations available in a given computational modelling environment. Even a 'short' implementation as this one requires a significant amount of time and intricate procedures to verify complex combinations such as the partitioned coupling FSI algorithm. As such, in this work particular combinations of the individual components making up the partitioned FSI algorithm are verified.

4.2 The Method of Manufactured Solutions

Verification and validation are essential procedures in computational science and engineering to certify a given modelling software implementation. In the field of computational mechanics validation has certainly become the standard procedure to attest to the 'correctness' of a model, while verification results are rarely presented. This could be due to several reasons, the most common being: lack of awareness of the need to verify the implementation, or restricted access to the source code.

The requirement to perform both certifying procedures can be best understood along the lines of simple yet powerful definitions/analogies provided by Patrick J. Roache. In [82] Roache describes verification as the process of providing evidence of the equations being solved correctly, whereas validation provides proof that the equations being solved are an appropriate representation of the physical phenomenon being considered. Or in Roache's words: '*verification* $\approx$ solving the equations right', '*validation* $\approx$ solving the right equations'.

The method of manufactured solutions (MMS) is a procedure to verify the accuracy of the solution produced by a code via systematic grid convergence testing, and it was originally implemented by Steinberg and Roache [83]. According to Roache [82] this procedure verifies the formulation of the discrete equations (interior and boundary conditions) and their order of accuracy, the accuracy of the solution procedure, and the user instructions. However, the method does not verify the coding accuracy of individual terms in mixed-order methods, nor does it verify coding mistakes that impact the iterative convergence rate of the solver decreasing its efficiency.

The definition of order of accuracy (p) given by Roache [82] relates the error of the discrete solution to a measure of the discretization (Δ) as:

$$e = f(\Delta) - f^{exact} = C\Delta^p + H.O.T. \tag{4.1}$$

with $H.O.T.$ as the collection of higher order terms. As seen in Eqn. (4.1), in order to evaluate the order of accuracy of the numerical implementation via systematic grid refinement the exact solution is required. Additionally, Eqn. (4.1) does not consider iteration errors

of the numerical solution, because of this, it is extremely important to set strict tolerances for the iterative solver so that its effect does not pollute the computation of the order of accuracy.

In recognition of the need to verify codes pertaining to general partial differential equations for which analytical solutions only exist for simplified cases (e.g. Navier-Stokes equations), the idea behind the MMS is to *manufacture* an exact solution to the problem which exercises all terms of the corresponding governing equation. Since verification is a *purely mathematical* exercise [82] the physical significance of the manufactured solution is of no importance. The end goal of the method is thus to provide a continuum analytical solution for which all the terms in the corresponding governing partial differential equation exist. The manufactured solution is then 'passed through the governing partial differential equation to give a production term $Q(x,y,z,t)$ that produces this solution' [82]. That is: given the partial differential equation (4.2)

$$\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} = 0 \tag{4.2}$$

an analytical solution $f(x,y)$ is generated and each term on the left hand side of the equation is evaluated accordingly to generate a source term $Q(x,y)$, such that $f(x,y)$ is the solution to

$$\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} = Q(x,y) \tag{4.3}$$

This source term is then added to the code to obtain a numerical solution for equation (4.3). This problem is then solved in several refined grids to compute the order of accuracy p of the solution.

The process of manufacturing an analytical solution is arguably not straightforward and guidelines can be found in [82] for that purpose. However, since the method has been used extensively to verify fluid dynamics problems abiding by the Naver-Stokes equations, there is a wide range of readily available manufactured solutions in the literature. Not having access to the source code was mentioned in the beginning of this chapter as one potential obstacle to perform code verification. This is a common situation

in commercial CFD codes where the user is limited to customise source terms only for the momentum governing equations. In recognition of this, several manufactured solutions which are divergence free have been devised. This project makes use of previously generated solutions for code verification.

Once this mathematical exercise has been completed the following things have been verified [82]:

- Any equation transformations (if non-orthogonal grids are used).
- The order of the discretization.
- The encoding of the discretization.
- The solution procedure.
- Inter-grid communication in overlapping zones.

Different types of boundary conditions can be tested with the same manufactured solution, however, the user needs to consider possible 'hard-coded' boundary values in the implementation. Such a situation is present in the original code developed by Skillen regarding the Neumann boundaries where the value is implicitly set to zero. Thus, in order to correctly test the Neumann conditions, the computational domain needs to be selected in a way that the normal gradient of the dependent variable at the boundary is effectively zero for the proposed manufactured solution. An alternative solution would be to generalise the code to accept any value for Neumann boundaries. This measure is not practical since all the Neumann boundaries for the problems treated in this thesis require a zero value. Throughout all cases presented in this chapter Dirichlet boundary conditions are used for the momentum equations, and a zero-pressure gradient in the boundary-normal direction is employed to solve the pressure-correction equation.

In this work, the Root Mean Square (RMS) error is used to quantify the inaccuracy introduced by the discretization practice in the momentum and mass conservation equations, and it is evaluated as:

$$e = \sqrt{\frac{\sum_n (u_n - U_n)^2 V_n}{\sum_n V_n}} \tag{4.4}$$

with u_n and U_n representing the numerical and manufactured solution for cell n respectively, with V as its volume. The observed order of accuracy p is computed from at least two solutions obtained on different grids as:

$$p \approx \frac{\log\left(\frac{e_{grid_1}}{e_{grid_2}}\right)}{\log r} \tag{4.5}$$

with grid 2 as the fine grid, and r the refinement ratio calculated as [82]:

$$r = \left(\frac{N_2}{N_1}\right)^{\frac{1}{D}} \tag{4.6}$$

where N_2 is the number of computational cells used for the fine grid and D being the dimensionality of the problem. Notice that it is not necessary to double the number of cells in each direction for the MMS verification procedure. However, it is recommended to do so in order to make the order-of-accuracy judgement easier [83].

This chapter makes extensive use of tables which contain all the relevant information about the verification procedure. It is therefore appropriate to explore the structure of such tables at this time. Referring to Table 4.1 from left to right: the first column indicates grid resoultion information by means of the number of cells in each direction. Since all cases presented in this Chapter are two-dimensional, if there is a third entry in the listing it is indicative of time-dependent simulations, and this entry represents the number of time steps used for time discretization. If there is a union symbol in this column it denotes the fact that overset grids have been used, indicating the background-regular grid resolutions. Columns 2 through 5 display information about the RMS error computed with Eqn. 4.4 for the x-momentum, y-momentum, the pressure field, and the continuity equation respectively. Similarly, columns 6 through 9 contain information about the observed order of accuracy (computed via Eqn. 4.5) between two consecutive grid refinements for x-momentum, y-momentum, the pressure field, and the continuity equation respectively.

4.3 Non-Overlapping Solver Verification

Steady State

The manufactured solutions shown in Eqns. (4.7)-(4.9)

$$u = \frac{1}{2}\cos(\pi\omega_0 x)\cos(\pi\omega_1 y) + \frac{1}{2} \tag{4.7}$$

$$v = \frac{1}{2}\sin(\pi\omega_0 x)\sin(\pi\omega_1 y) + \frac{1}{2} \tag{4.8}$$

$$p = \cos(\pi\omega_0 x)\cos(\pi\omega_1 y) + \frac{1}{2} \tag{4.9}$$

are modified forms of the ones introduced in [84] for a time-dependent case, and they are used here to verify the non-overlapping steady state solver. The solution is divergence free for $\omega_0 = \omega_1$. For all the remaining exercises of this chapter, verification results are presented using the QUICK convection discretization scheme and two-dimensional cases, as those conditions were employed in all FSI validation instances. The base computational domain is a square cavity with 10×8 cells as seen in Fig. 4.1, and the solution is computed in four successive grid refinements of factor 2 each. When employing the MMS it is recommended to set up the problem with non-symmetrical grid spacing along the $x - y$ directions to expose possible encoding errors that could potentially be undetected by the use of symmetrical discretization. Dirichlet boundary conditions are employed along all boundaries for the momentum equations, whereas the dimensions of the cavity $((-2 \times 2), (-2 \times 2))$ are selected so that the pressure gradient normal to the four boundaries evaluates to zero to abide by the zero-pressure gradient condition hard-coded by Skillen. RMS error behaviour is presented graphically in Fig. 4.2 where it can be seen that the continuity RMS error decreases at a faster rate than the momentum and pressure errors as the grid is refined, a sample second order slope has been included in the plot for comparison purposes. The computed observed order is shown in Table 4.1. Results agree with the theoretical expected second order of accuracy.

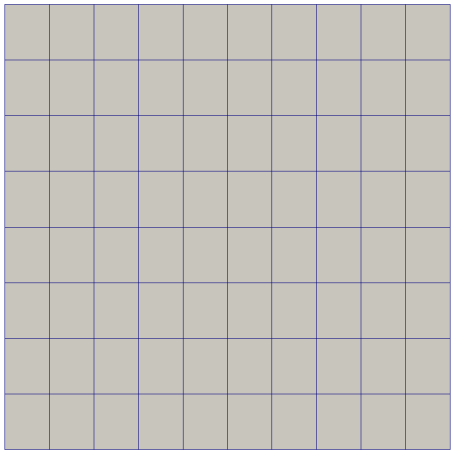

Figure 4.1: Base grid employed for non-overlapping steady state verification.

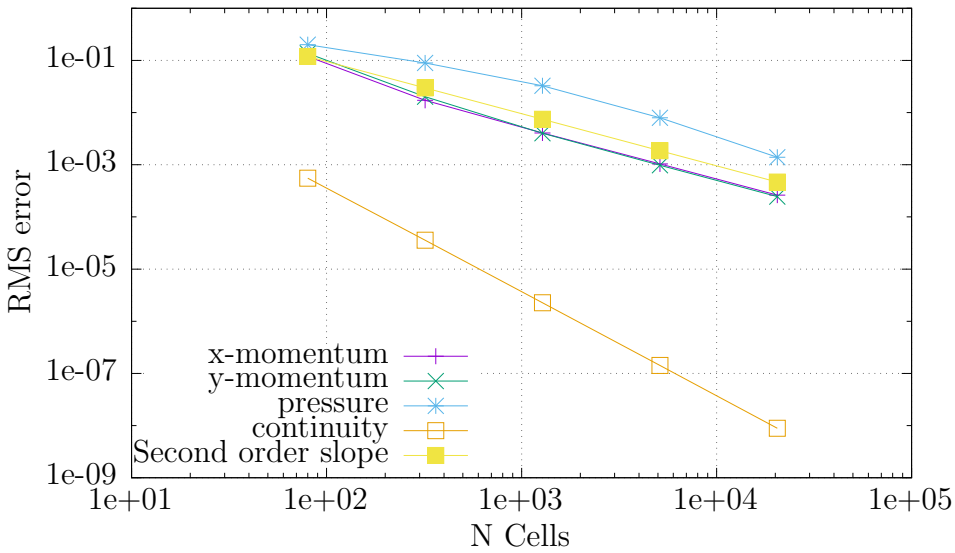

Figure 4.2: RMS error, steady, non-overlapping.

Transient first and second order solutions

The original manufactured solution presented in [84] is now employed as a benchmark for all time-dependent cases:

$$u = \frac{1}{2}\cos(\pi\omega_0 x)\cos(\pi\omega_1 y)\cos(\pi t) + \frac{1}{2} \tag{4.10}$$

$$v = \frac{1}{2}\sin(\pi\omega_0 x)\sin(\pi\omega_1 y)\cos(\pi t) + \frac{1}{2} \tag{4.11}$$

$$p = \cos(\pi\omega_0 x)\cos(\pi\omega_1 y)\cos(\pi t) + \frac{1}{2} \tag{4.12}$$

It includes a $\cos(\pi t)$ factor for momentum and pressure. For time-dependent problems this solution is evaluated at $t = 0$ and the resulting fields are used as initial conditions. However, this practice is not recommended as some coding errors can slip past the verification test [85]. Instead, recommended initialisation procedures are to use the exact field multiplied by a constant, or to employ the *numerical* steady state solution if possible. The latter course of action was taken in this work. Time-dependent Dirichlet boundary conditions are used on all domain bounding faces, along with a zero-pressure gradient in the surface normal direction.

The computational domain is exactly the same as for the steady state case, with grid refinement ratios of 2.519 and 2 for the first order Euler time scheme, and the second order Backward time scheme respectively. The 2.519 refinement ratio for the Euler scheme is the result of an attempt to keep *both* time (first order) and spatial (second order) error reduction ratios the same for the verification procedure [86].

The grid description column in Tables 4.2 and 4.3 includes the number of time steps needed to reach the desired transient solution at $t = 1s$ as the last number. As shown in Table 4.2 the order of accuracy of the solution when using the Euler first scheme is inferior to the Backward scheme, converging approximately to a value of 1.4 for the momentum equations. The observed order of the solution employing the Backward scheme is effectively 2 for the momentum equations as seen in Table 4.3.

Grid	u_{err}	v_{err}	p_{err}	$(\nabla \cdot \mathbf{U})_{err}$	Obs. u	Obs. v	Obs. p	Obs. $(\nabla \cdot \mathbf{U})$
10×8	1.1924e-01	1.3707e-01	2.0058e-01	5.52496e-04	—	—	—	—
20×16	1.7253e-02	2.0135e-02	8.9721e-02	3.58781e-05	2.788	2.767	1.160	3.944
40×32	4.0608e-03	4.0202e-03	3.2602e-02	2.2638e-06	2.087	2.324	1.460	3.986
80×64	1.0407e-03	9.7782e-04	7.9584e-03	1.41823e-07	1.964	2.039	2.034	3.996
160×128	2.6247e-04	2.4553e-04	1.3892e-03	8.86921e-09	1.987	1.993	2.518	3.999

Table 4.1: Order of accuracy, non-overlapping grids, steady state, ref. ratio 2.

Grid	u_{err}	v_{err}	p_{err}	$(\nabla \cdot \mathbf{U})_{err}$	Obs. u	Obs. v	Obs. p	Obs. $(\nabla \cdot \mathbf{U})$
$10\times8\times8$	7.56311e-02	7.24573e-02	3.31978e-01	5.52496e-04	—	—	—	—
$20\times16\times32$	1.6785e-02	9.57784e-03	1.4442e-01	3.58781e-05	1.628	2.189	0.900	2.958
$40\times32\times128$	4.34017e-03	2.40763e-03	4.18536e-02	2.2638e-06	1.463	1.494	1.340	2.989
$80\times64\times512$	1.10141e-03	6.40091e-04	9.50818e-03	1.41823e-07	1.483	1.433	1.603	2.997
$160\times128\times2048$	2.7652e-04	1.63515e-04	1.64304e-03	8.86932e-09	1.495	1.477	1.900	3.000

Table 4.2: Order of accuracy, for non-overlapping, transient Euler scheme, ref. ratio 2.519.

Grid	u_{err}	v_{err}	p_{err}	$(\nabla \cdot \mathbf{U})_{err}$	Obs. u	Obs. v	Obs. p	Obs. $(\nabla \cdot \mathbf{U})$
$10\times8\times8$	6.2283e-02	6.26438e-02	3.18421e-01	5.52496e-04	—	—	—	—
$20\times16\times16$	1.35232e-02	9.81321e-03	1.17439e-01	3.58781e-05	2.203	2.674	1.439	3.944
$40\times32\times32$	3.40217e-03	2.56977e-03	3.44256e-02	2.2638e-06	1.990	1.933	1.770	3.986
$80\times64\times64$	8.62784e-04	6.89122e-04	8.0754e-03	1.41823e-07	1.979	1.898	2.091	3.996
$160\times128\times128$	2.17217e-04	1.77339e-04	1.42092e-03	8.86989e-09	1.989	1.958	2.506	3.999

Table 4.3: Order of accuracy, for non-overlapping, transient Backward scheme, ref. ratio 2.

4.4 Overset Solver Verification

Steady state

The next logical step in the verification procedure is to quantify the effect of inter-grid communication on the order of accuracy for overlapping grids. With the steady state manufactured solution presented in section 4.3, two cases involving the use of an overset mesh are presented using either standard linear interpolation, or mass flux based interpolation in the overlapping zone. The square cavity is now employed as the background grid, with a rotated overlapping square grid added to the computational domain as shown in Fig. 4.3. Boundary conditions remain the same as in the non-overlapping verification case for the background grid, and two-way inter-grid communication is enabled in the overlapping zone.

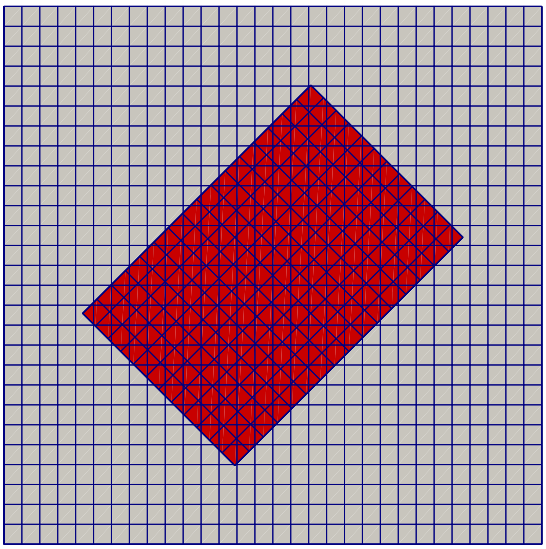

Figure 4.3: Overset grid base configuration for inter-grid communication verification.

Three successive grid refinements of ratio 2 are performed on the base configuration depicted in Fig 4.3. Results (Tables 4.4, 4.5) show a clear decrease in the order of accuracy when compared to the non-overlapping solution. A graphical comparison of the RMS error behaviour between the use of overlapping and non-overlapping grids (Fig. 4.4) confirms this fact, as the slope of the error when using overset grids is less pronounced than the slope of the error for non-overlapping grids. The order of accuracy of the method when employing the mass flux based interpolation seems to converge faster to a value of

1 than it does for the case of standard linear interpolation, especially for the momentum equations. The computed horizontal velocity fields (U) using MFBI for the coarse and fine grids are presented in Figure 4.6, and Figure 4.7 respectively.

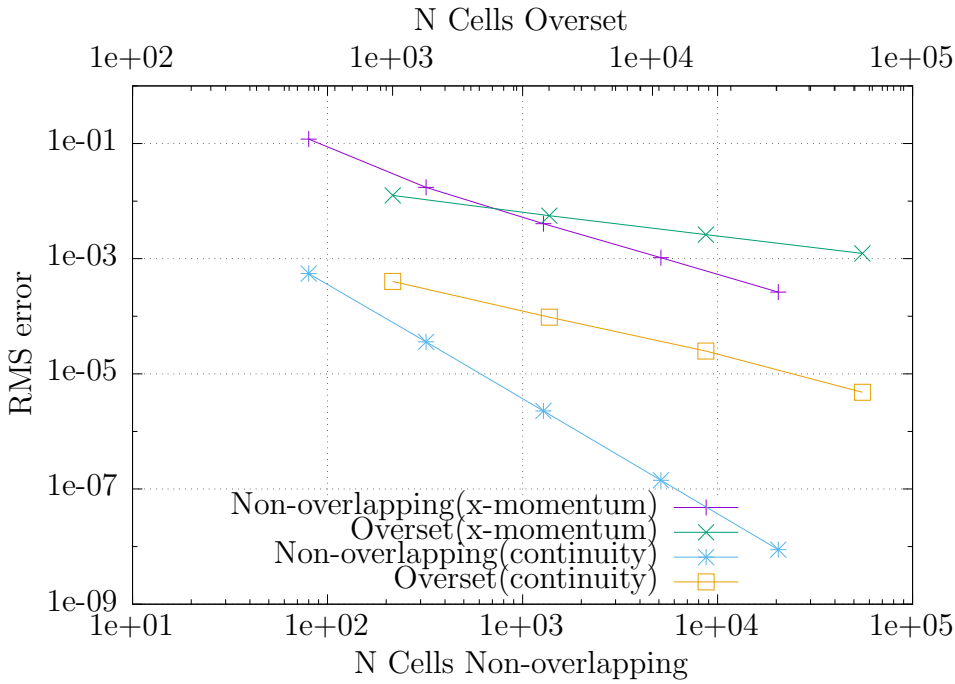

Figure 4.4: RMS error comparison, MFBI vs Non-overlapping.

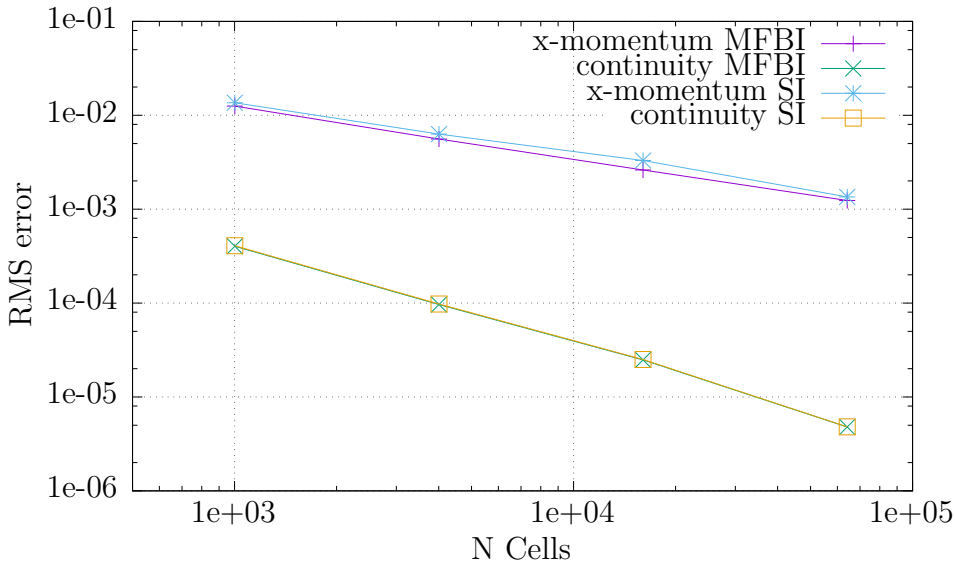

Figure 4.5: Accuracy comparison. MFBI vs SI.

Another interesting result is the comparison between the accuracy of MFBI against its SI counterpart. Figure 4.5 formally confirms that, as expected, MFBI is generally more accurate for a given overset case setup than SI for momentum, whilst the difference for mass conservation is negligible.

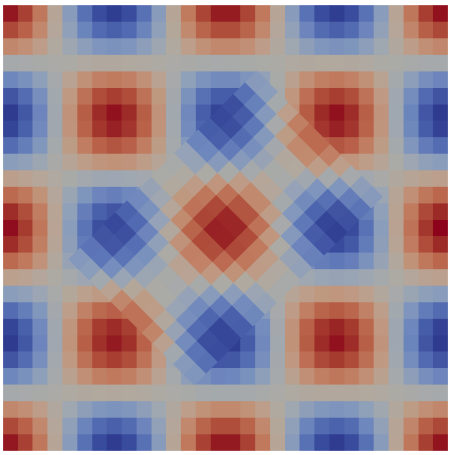

Figure 4.6: U_x velocity field. Overset Coarse grids.

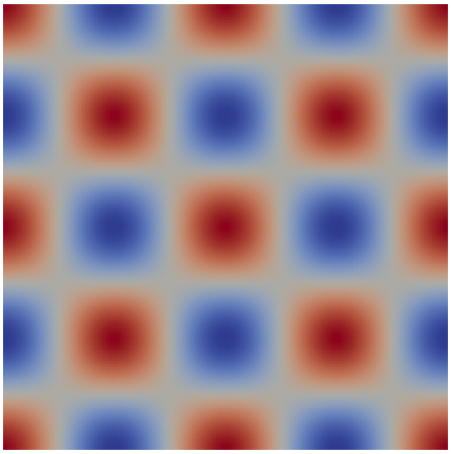

Figure 4.7: U_x velocity field. Overset Fine grids.

Grid	u_{err}	v_{err}	p_{err}	$(\nabla \cdot \mathbf{U})_{err}$	Obs. u	Obs. v	Obs. p	Obs. $(\nabla \cdot \mathbf{U})$
$30\times27\cup16\times12$	1.36054e-02	1.03229e-02	4.90913e-02	4.08437e-04	—	—	—	—
$60\times54\cup32\times24$	6.2952e-03	5.70726e-03	1.27023e-02	9.78482e-05	1.111	0.854	1.950	2.061
$120\times108\cup64\times48$	3.30055e-03	3.2092e-03	4.03809e-03	2.50598e-05	0.931	0.830	1.653	1.965
$240\times216\cup128\times96$	1.35257e-03	1.48027e-03	2.10391e-03	4.80501e-06	1.287	1.116	0.940	2.382

Table 4.4: Order of accuracy,overlapping, steady, linear interpolation ref. ratio 2.

Grid	u_{err}	v_{err}	p_{err}	$(\nabla \cdot \mathbf{U})_{err}$	Obs. u	Obs. v	Obs. p	Obs. $(\nabla \cdot \mathbf{U})$
$30\times27\cup16\times12$	1.25395e-02	9.73036e-03	4.75059e-02	4.024e-04	—	—	—	—
$60\times54\cup32\times24$	5.59157e-03	4.79258e-03	1.62656e-02	9.64556e-05	1.165	1.021	1.546	2.060
$120\times108\cup64\times48$	2.61341e-03	2.43348e-03	4.35762e-03	2.48594e-05	1.097	0.977	1.900	1.956
$240\times216\cup128\times96$	1.23578e-03	1.21618e-03	1.57413e-03	4.78758e-06	1.080	1.000	1.474	2.376

Table 4.5: Order of accuracy,overlapping, steady, MFBI interpolation ref. ratio 2.

Transient

The results of two additional verification exercises involving time-dependent flow, along with static overlapping grids are presented in Tables 4.6, and 4.7. Linear inter-grid interpolation was used for this test, and the solution computed at $t = 1s$. In both instances the error behaviour for the last grid refinement deviates from the first three solutions. This may be due to the introduction of iteration errors in the error computation, as for the last grid resolution the convergence criterion at each time step had to be relaxed considerably in order to prevent excessively long simulation running times.

Grid	u_{err}	v_{err}	p_{err}	$(\nabla \cdot \mathbf{U})_{err}$	Obs. u	Obs. v	Obs. p	Obs. $(\nabla \cdot \mathbf{U})$
$30\times27\cup16\times12\times5$	4.20188e-02	3.46593e-02	2.223371e-01	9.03738e-04	—	—	—	—
$60\times54\cup32\times24\times20$	1.29919e-02	1.0632e-02	6.5625e-02	2.50284e-04	1.270	1.279	1.320	1.389
$120\times108\cup64\times48\times80$	3.80182e-03	3.43663e-03	1.90951e-02	6.86433e-05	1.330	1.222	1.336	1.400
$240\times216\cup128\times96\times320$	1.66129e-03	1.30259e-03	5.43443e-03	1.5154e-05	0.896	1.050	1.360	1.635

Table 4.6: Order of accuracy,overlapping, transient, linear interpolation, Euler, ref. ratio 2.519.

Grid	u_{err}	v_{err}	p_{err}	$(\nabla \cdot \mathbf{U})_{err}$	Obs. u	Obs. v	Obs. p	Obs. $(\nabla \cdot \mathbf{U})$
$30\times27\cup16\times12\times5$	2.72743e-02	2.04493e-02	9.3422e-02	9.17359e-04	—	—	—	—
$60\times54\cup32\times24\times10$	8.68613e-03	6.64363e-03	2.33463e-02	2.51953e-04	1.650	1.622	2.000	1.864
$120\times108\cup64\times48\times20$	2.73509e-03	2.54979e-03	8.22756e-03	6.87954e-05	1.667	1.381	1.504	1.872
$240\times216\cup128\times96\times40$	1.45892e-03	1.10819e-03	3.19736e-03	1.51641e-05	0.906	1.202	1.363	2.181

Table 4.7: Order of accuracy,overlapping, transient, linear interpolation, Backward ref. ratio 2.

4.5 Space Conservation Law and ALE Verification

The traditional approach to verify a Space Conservation Law (SCL) implementation entails the computation of a trivial solution for the flow and pressure fields as grid motion takes place. This approach shall be used in this section to verify the implementation of the SCL under grid deformation. However, the MMS still proves to be a valuable tool for this exercise under the realisation that any solution to the governing equations should remain undisturbed solely by grid motion. As such, the time-dependent manufactured solution introduced in section 4.3 is used to verify the SCL under translational and rotational overset motion. The same set of grids employed for steady state overset verification are used for the non-deforming cases.

For all overset grid motion cases the solution is obtained with the two available flow time discretization schemes along with linear inter-grid interpolation. This is done because the discretization of the SCL is different for first and second order time schemes. Besides serving as formal verification of the SCL, the results presented in this section also attest for the mathematical correctness of a new combination of solver settings: time-dependent problems using moving overlapping grids, which implicitly also involves the ALE formulation.

Overset Translational Motion.

The initial overset grid position and U velocity field is shown in Fig. 4.8, with the corresponding steady state *numerical* solution serving as initial condition for each grid refinement. The solution is then calculated at $t = 1s$ with the number of time steps required shown as the last number in the Grid column on Tables 4.8 and 4.9. The overset grid is then subject to the same constant velocity in the horizontal and vertical directions ($0.50\frac{m}{s}$). Figue 4.9 shows the final position of the overset grid along with the computed U velocity field using the first order time scheme for an intermediate grid in the refinement sequence. The grid refinement ratio for the first order implementation is different from the one for the second order implementation in order to keep the error reduction ratio the same for spatial and time discretization as was done in section 4.3.

The overall resulting order of accuracy for the Backward time scheme case is higher

than for the Euler scheme (see Tables 4.8, 4.9), which is actually higher than 1 and converging towards 1.3 for the momentum equations. It is interesting to note that the difference between the order of accuracy of both schemes under these conditions is less ($\approx 20\%$ vs $\approx 25\%$ for the momentum equations) than what it was for the time-dependent non-overlapping cases.

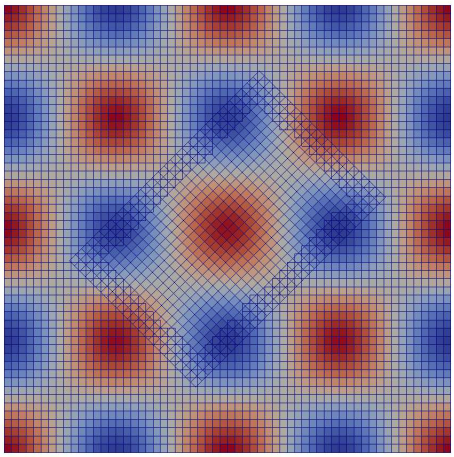

Figure 4.8: Initial U_x field and grids for the translational motion verification exercise.

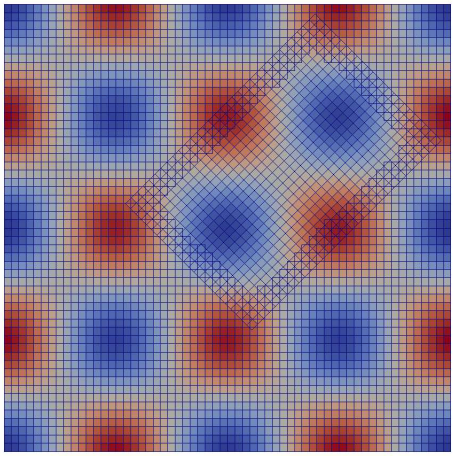

Figure 4.9: Computed U_x field and grids for the translational motion verification exercise.

Grid	u_{err}	v_{err}	p_{err}	$(\nabla \cdot \mathbf{U})_{err}$	Obs. u	Obs. v	Obs. p	Obs. $(\nabla \cdot \mathbf{U})$
$30\times27\cup16\times12\times5$	4.24222e-02	3.52758e-02	2.13132e-01	1.01947e-03	—	—	—	—
$60\times54\cup32\times24\times20$	1.27483e-02	1.05996e-02	6.129e-02	2.56512e-04	1.301	1.301	1.349	1.493
$120\times108\cup64\times48\times80$	3.82314e-03	3.23518e-03	1.56232e-02	6.40608e-05	1.303	1.284	1.479	1.501

Table 4.8: Order of accuracy, space conservation law, translation Euler time scheme, ref. ratio 2.519.

Grid	u_{err}	v_{err}	p_{err}	$(\nabla \cdot \mathbf{U})_{err}$	Obs. u	Obs. v	Obs. p	Obs. $(\nabla \cdot \mathbf{U})$
$30\times27\cup16\times12\times5$	2.73313e-02	2.12092e-02	8.60406e-02	1.03319e-03	—	—	—	—
$60\times54\cup32\times24\times10$	8.51585e-03	6.61382e-03	2.06392e-02	2.58548e-04	1.682	1.681	2.059	1.998
$120\times108\cup64\times48\times20$	2.77795e-03	2.30333e-03	5.69377e-03	6.42614e-05	1.616	1.521	1.857	2.008

Table 4.9: Order of accuracy, space conservation law, translation Backward time scheme, ref. ratio 2.

Grid	u_{err}	v_{err}	p_{err}	$(\nabla \cdot \mathbf{U})_{err}$	Obs. u	Obs. v	Obs. p	Obs. $(\nabla \cdot \mathbf{U})$
$30\times27\cup16\times12\times5$	4.20568e-02	3.50849e-02	2.25218e-01	8.04296e-04	—	—	—	—
$60\times54\cup32\times24\times20$	1.30339e-02	1.07463e-02	7.13758e-02	2.04781e-04	1.268	1.280	1.243	1.480
$120\times108\cup64\times48\times80$	3.82613e-03	3.48335e-03	2.35199e-02	5.21122e-05	1.326	1.219	1.201	1.481

Table 4.10: Order of accuracy, space conservation law, Euler time scheme rotation case, ref. ratio 2.519.

Grid	u_{err}	v_{err}	p_{err}	$(\nabla \cdot \mathbf{U})_{err}$	Obs. u	Obs. v	Obs. p	Obs. $(\nabla \cdot \mathbf{U})$
$30\times27\cup16\times12\times5$	2.74076e-02	2.11036e-02	9.50838e-02	8.0578e-04	—	—	—	—
$60\times54\cup32\times24\times10$	8.59874e-03	6.77902e-03	3.0397e-02	2.05834e-04	1.672	1.638	1.645	1.972
$120\times108\cup64\times48\times20$	2.77438e-03	2.68124e-03	1.50576e-02	5.21776e-05	1.631	1.338	1.013	1.976

Table 4.11: Order of accuracy, space conservation law, Backward time scheme rotation case, ref. ratio 2.

Overset Rotational Motion.

The initial position of the overlapping grid is the same as it was for the previous case, and it is presented in Figure 4.10 along with the initial numerical pressure field. The overset grid has a constant angular velocity of $0.50\frac{rad}{s}$ and the solution is calculated at time $t = 1s$. Figure 4.11 shows the final overset grid orientation and the resulting pressure field obtained using the Backward time discretization scheme.

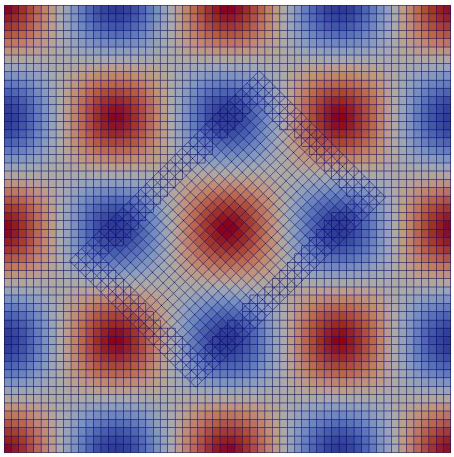

Figure 4.10: Initial pressure field and grids for rotational motion verification exercise.

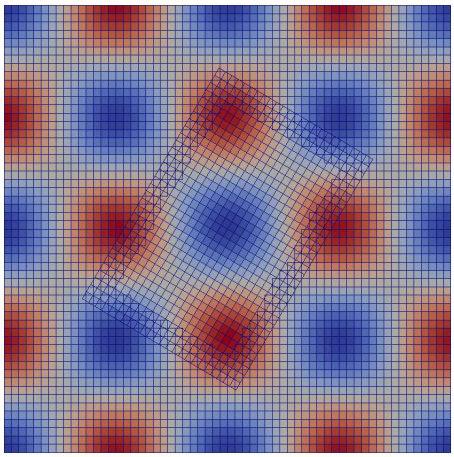

Figure 4.11: Final pressure field and grids for rotational motion verification exercise.

The resulting order of accuracy for both time discretization schemes (Tables 4.10, 4.11) is quite similar to the translational motion case. This is hardly a surprising result, since the only variant introduced is the type of oveset grid motion which ultimately should not have any effect on the computed fields.

Grid deformation.

Even though *rigid-body* FSI is the subject of this thesis, the SCL is subject to deformation tests to further verify its implementation. The tests presented in this section employ an initial trivial solution field ($\mathbf{U} = 0, P = 0$) to demonstrate the enforcement of the SCL under grid distortion.

The first case makes use of an overset inner grid enclosed in a square cavity (Fig. 4.12) whose bounding sides are static walls. The overset grid is then allowed to undergo deformation for 500 time steps of 5 *ms* each using the Euler first order time discretization scheme. A successful implementation of the space conservation law should recover the initial static field, since no external interactions were applied through the domain boundaries. This is certainly the case in Fig. 4.13 as the velocity magnitude field is practically zero at the end of the simulation. In order to visualize the effects of an incorrect implementation, an incorrect sign was purposely introduced in the mesh flux term balance in the discretized pressure correction equation for moving grids. The resulting spurious velocity magnitude field is no longer at rest as shown in Fig. 4.14. Another useful measure of the degree of compliance of the SCL is the *local continuity error*, which is the absolute difference between the change in volume of a computational cell and the sum of the swept volumes by all faces in the cell during a given time step. The reported error is the weighted average of errors from all the cells which are weighted using their corresponding volumes [76]. Figs. 4.15, 4.16 show the local continuity errors for the enforcement of the SCL, and for the violation of the SCL respectively. The local continuity error for the non-SCL compliant case is about twelve orders of magnitude higher than its SCL enforcing counterpart.

Finally, the same static field is used to analyse the SCL compliance when the Backward second order accurate time scheme is selected. For this test only the main square

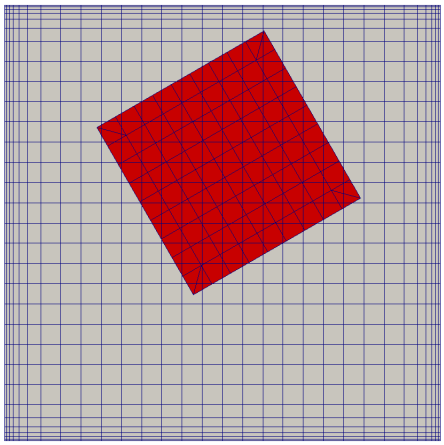

Figure 4.12: Grid deformation, Euler SCL, initial state.

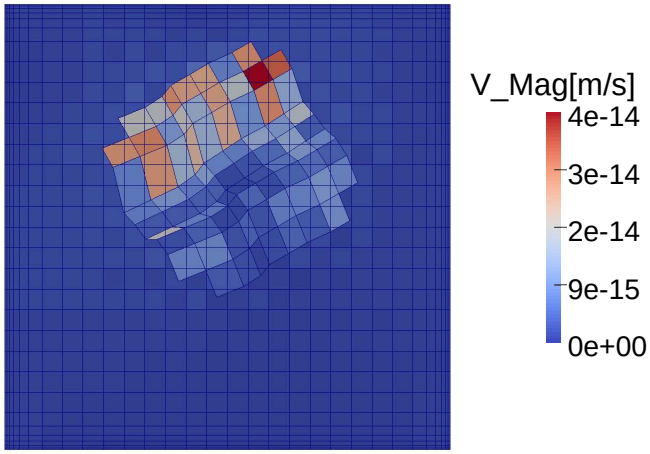

Figure 4.13: Euler first order, final state, SCL enforced.

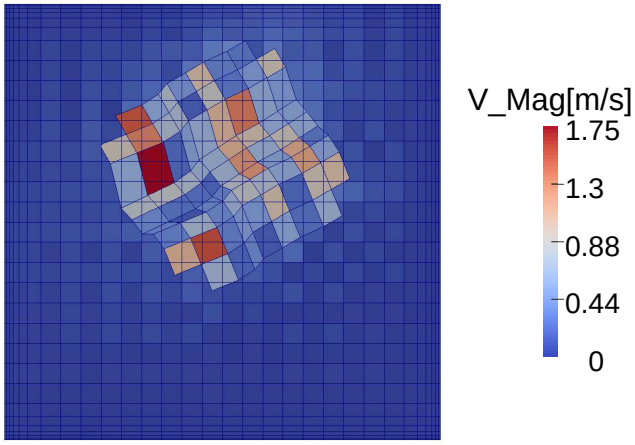

Figure 4.14: Euler first order, final state, SCL not enforced.

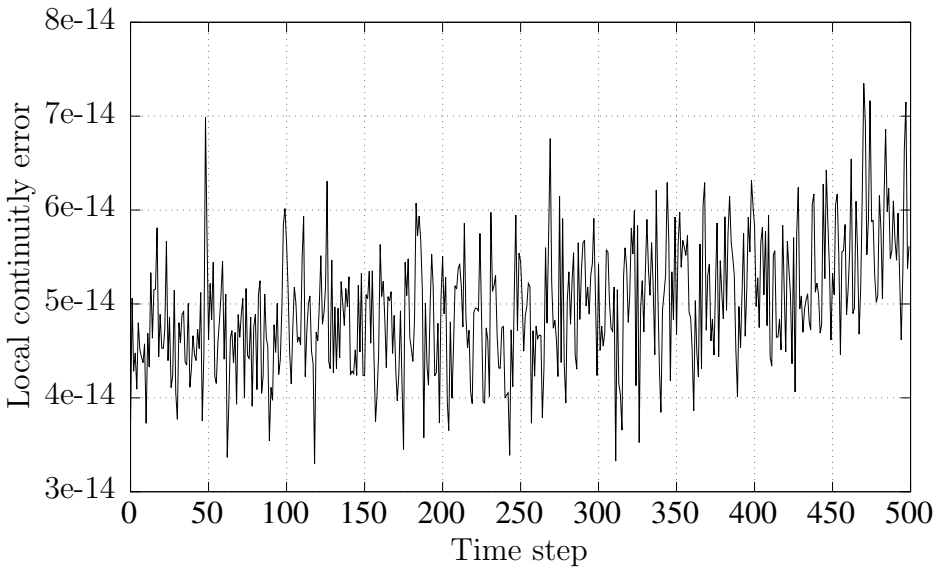

Figure 4.15: Continuity errors, SCL enforced.

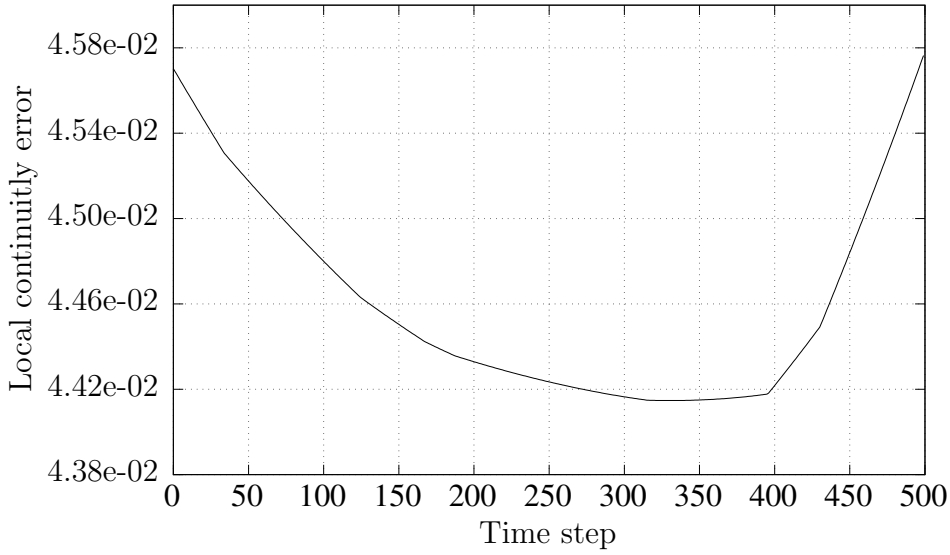

Figure 4.16: Continuity errors, SCL not enforced.

cavity grid is present, and deformed over the course of 125 time steps. The velocity magnitude field is SCL-compliant as visualised through Fig. 4.17, and the corresponding local continuity error is presented in Fig. 4.18.

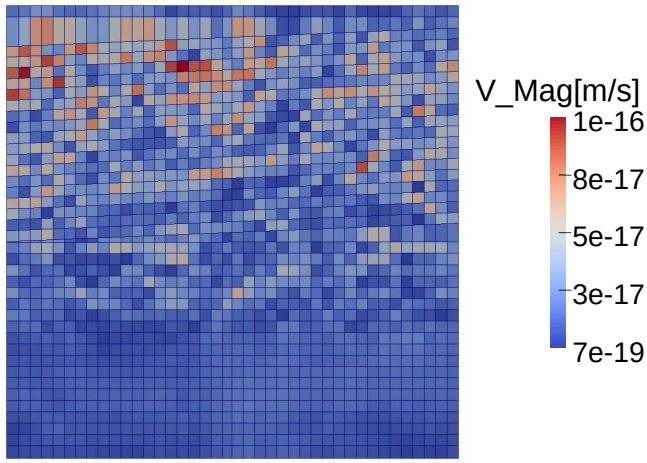

Figure 4.17: Backward second order, final state, SCL enforced.

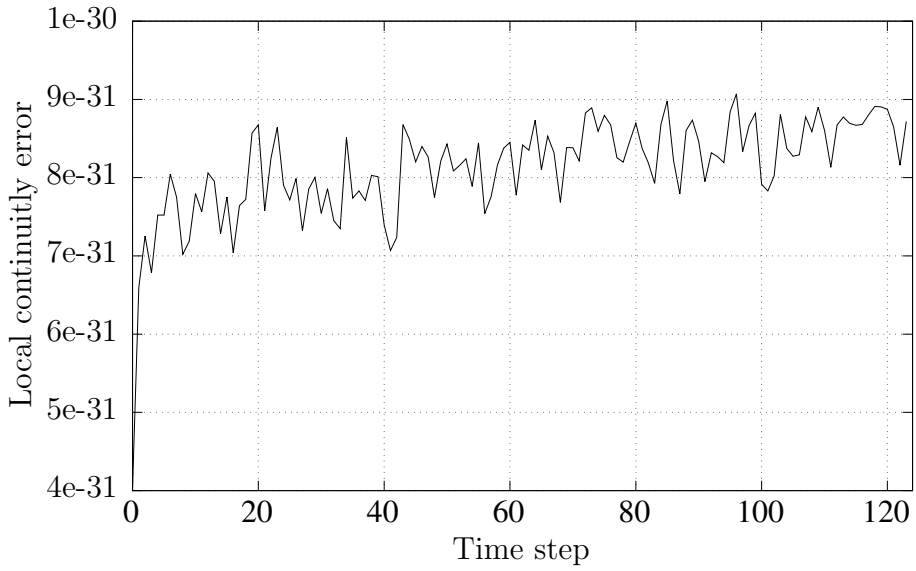

Figure 4.18: Backward, continuity errors, SCL enforced.

4.6 Rigid-Body Solver Verification

Two cases are presented here to verify the implementation of the rigid-body motion solver. The first case pertains to the solution of one dimensional translational motion, whereas the second case models rigid-body rotation. Both cases are solved using the second order Heun method, and the third order Adams-Moulton, Adams-Bashforth method introduced in section 3.3.5. Whenever the third order method is used, Heun's method serves as a 'starter' to provide two previous time step solutions required for the Adams method.

Analytical solutions for the differential equations are readily available, and are used to compute the RMS error, and the observed order of accuracy of the numerical methods. Since the computational domain for this rigid-body solver verification exercise comprises uniform 'time' control volumes, the RMS error is computed as:

$$e_2 = \sqrt{\frac{1}{N}\sum_n (u_n - U_n)^2} \tag{4.13}$$

with N the number of time steps, u_n the numerical solution, and U_n the analytical solution. The observed order is computed using Eqn. 4.5.

Translational motion

A free vibrating spring-mass system of mass $m = 1\ kg$, and spring stiffness $k = 10\ \frac{N}{m}$ is allowed to undergo translational motion in one direction, and is modelled through Newton's second law. The resulting governing differential equation is:

$$m\frac{d^2x}{dt^2} + kx = 0 \tag{4.14}$$

subject to $x(0) = 0$, $x'(0) = \frac{1}{2}\ \frac{m}{s}$ as initial conditions. The system is in motion for 5 s, with the analytical solution given by:

$$x(t) = \frac{\sin(\sqrt{10}t)}{2\sqrt{10}} \tag{4.15}$$

Tables 4.12, and 4.13 confirm the expected results, i.e. second-order accuracy for Heun's method, and third-order accuracy for the Adams method. Fig 4.19 is a graphical comparison of the RMS error behaviour as the computational grid is refined, whilst Fig. 4.20 is an actual comparison of the numerical solution obtained with 80 time steps using the Adams method, and the analytical solution.

N. steps	RMS_{err}	Obs. order
20	5.66621e-02	—
40	1.34412e-02	2.0757
80	3.37797e-03	1.992
160	8.44298e-04	2.000

Table 4.12: Observed order of accuracy. Translational motion. Heun's method.

N. steps	RMS_{err}	Obs. order
20	2.50174e-02	—
40	2.71577e-03	3.203
80	3.357e-04	3.016
160	4.19478e-05	3.0005

Table 4.13: Observed order of accuracy. Translational motion. Adams' method.

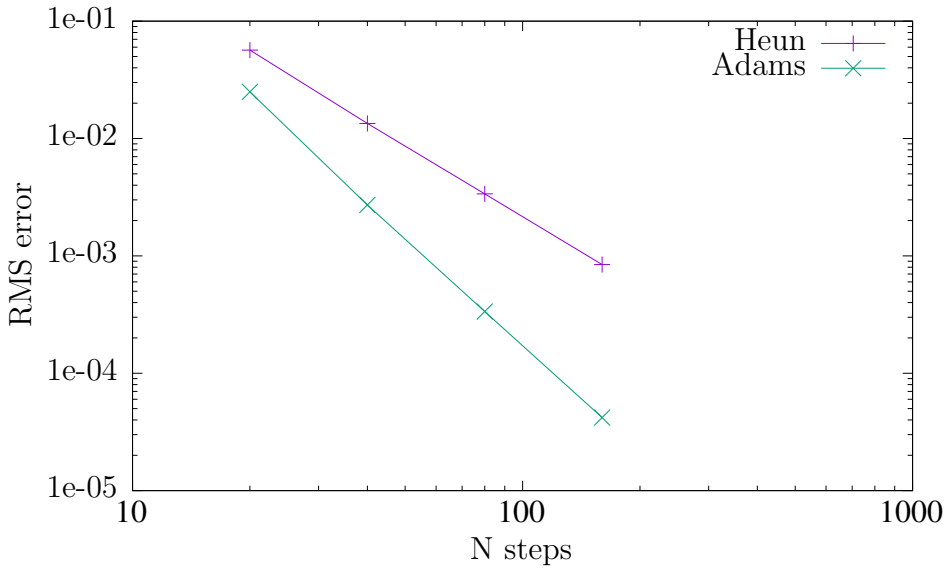

Figure 4.19: Rigid-body solver verification. Translational motion.

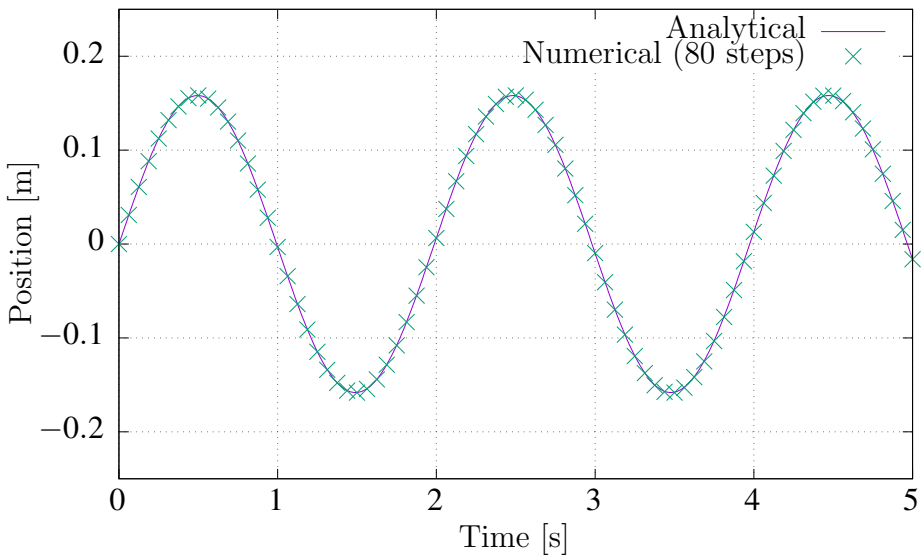

Figure 4.20: Analytical vs Numerical solution (Adams). Translational motion.

Rotational motion

The rotational motion verification exercise also serves to corroborate the implementation of quaternions as parameters to populate the rotation matrix, which ultimately represents the orientation of the rigid body relative to the inertial reference frame.

In this case, a mass of inertia $I = 1 \times 10^{-3}\ kg \cdot m^2$ is allowed to experience rotational motion about the z axis. The mass is subject to the action of a rotational damper whose damping constant is $B = 1 \times 10^{-2}\ \frac{N \cdot m \cdot s}{rad}$, and a rotational spring with constant $K = 9 \times 10^{-1}\ \frac{N \cdot m}{rad}$. The initial conditions for the system are: $\theta(0) = 0\ rad$, and $\theta'(0) = 0.50\ \frac{rad}{s}$.

The governing differential equation is:

$$I\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} + K\theta = 0 \tag{4.16}$$

with its corresponding analytical solution being:

$$\theta(t) = 1.69031 \times 10^{-02} \exp(-5t)\sin(5\sqrt{35}t) \tag{4.17}$$

The observed order of accuracy after the first grid refinement is surprisingly high for both methods as seen in Tables. 4.14, and 4.15. This unstable error behaviour is also experienced sometimes during the first grid refinements presented throughout this chapter for the flow solver, and is most likely associated with the fact that the leader order error in a Taylor series truncation is not actually significantly larger than the higher order terms for the initial coarse grids. However, as the grid is further refined, the error behaviour stabilises as reflected by convergence of the order of accuracy to the expected values, and supported by constant slopes in Fig 4.21. The numerical solution of the differential equation for a coarse grid comprising 40 time steps for a total simulation duration of $1s$ is presented in Fig. 4.22 exhibiting reasonable agreement with the analytical solution.

N. steps	RMS_{err}	Obs. order
20	4.69063e-02	—
40	7.65519e-04	5.937
80	1.86532e-04	2.037
160	4.68313e-05	1.993

Table 4.14: Observed order of accuracy. Rotational motion. Heun's method.

N. steps	RMS_{err}	Obs. order
20	2.41662e-02	—
40	3.16113e-04	6.256
80	3.70352e-05	3.0934
160	4.60685e-06	3.00704

Table 4.15: Observed order of accuracy. Rotational motion. Adams' method.

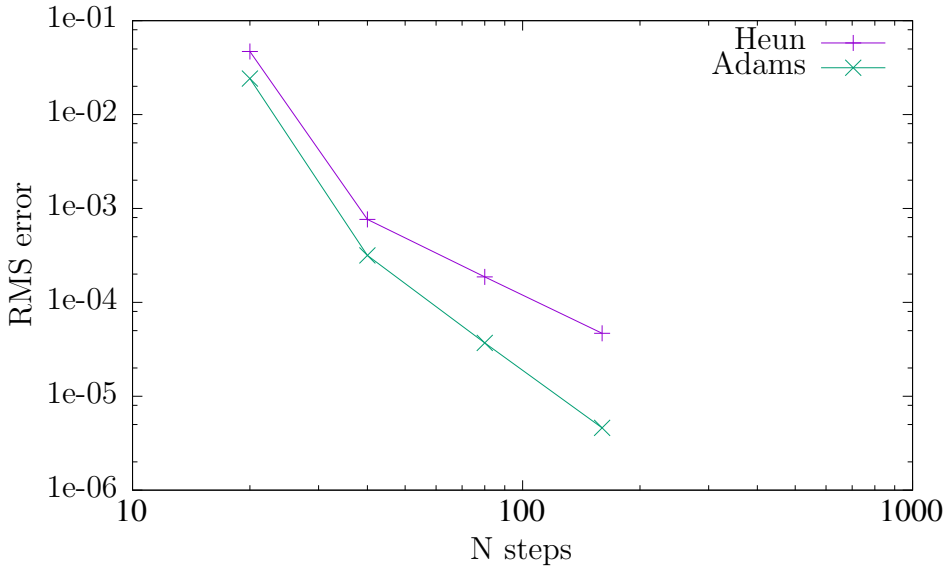

Figure 4.21: Rigid-body solver verification. Rotational motion.

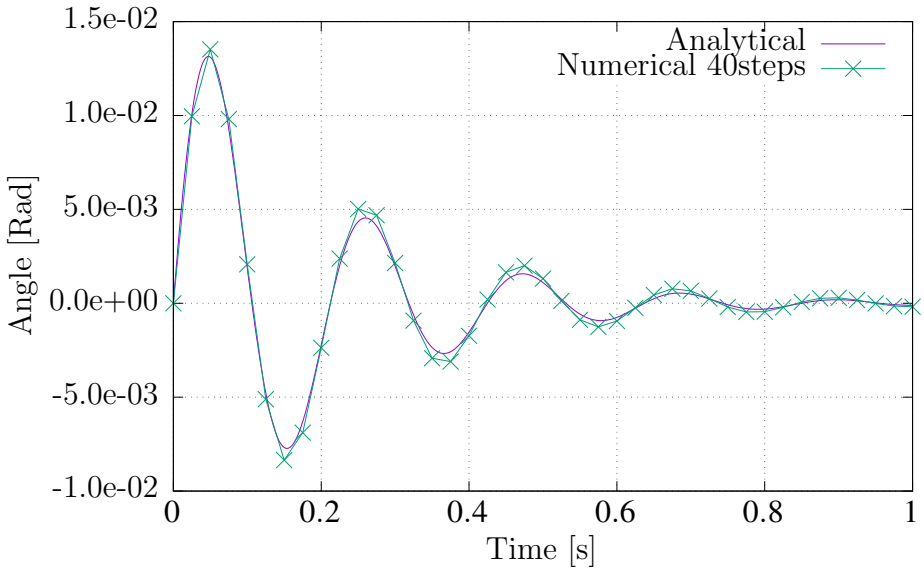

Figure 4.22: Analytical vs Numerical solution (Adams). Rotational motion.

4.7 Closure

The ability of this numerical modelling environment to produce mathematically correct solutions has been demonstrated via the MMS for the fluid solver, and the method of exact solutions (MES) for the rigid-body solver. As a by-product of this exercise, a solid estimate of the order of accuracy of the computational model under different circumstances has been obtained, showing good agreement with the expected theoretical values.

Some of the most significant results in terms of accuracy assessment are: overset grids decrease the order of accuracy of the numerical solution in this implementation, both inter-grid communication procedures exhibit virtually the same error reduction rate as the computational domain is refined, the difference between the second order Backward scheme and the first order Euler scheme regarding the error reduction rate decreases with the use of overlapping grids, and the introduction of the ALE formulation along with the SCL for moving grids does not further degrade the order of accuracy when using overlapping grids.

The implementation of the fundamental components for a grid motion formulation, namely the SCL and the ALE form of the governing equations, have been subject to different motion conditions always providing satisfactory evidence of their correctness.

Chapter 5

One-Way FSI Simulations

5.1 Introduction

As seen in the previous Chapter, the individual components of the overset time-dependent FSI implementation have been tested and formally proved to produce mathematically correct solutions. The next step is to examine the coupling of such elements to solve problems in which fluid immersed rigid-body objects with prescribed motion interact with the flow field, namely One-Way FSI simulations.

Since the main objective of this Chapter is to validate the One-Way FSI algorithm, grid independent results were not a main concern. Still, all the problems presented here have been solved using at least two grid resolutions.

As seen in the literature review section of this thesis, even though the overset grid technique has been around for at least about 30+ years, its availability in commercial or Open-source codes has been limited for the general public. CD-adapco was perhaps the first company to introduce it to its users (around 2011-2013). Recently (2016), ANSYS-Fluent introduced the feature in version 17.0 but was limited to static grids, and later expanded it to support grid motion in version 18.0.

Within the Open-source community the overset technique was made available through licensed third party libraries since at least 2010 (SUGGAR++), and had to be manually coupled to CFD solvers such as OpenFOAM® . As per licensing requirements these libraries had a cost and/or in some instances their use was even limited to users from certain countries (for example NASA'S OVERFLOW solver).

In January 2017 the development group led by H. Jasak made available for the first time a truly Open-source overset implementation compatible with FOAM-extend 4.0. Still, the code has to be requested and compiled manually by the user. In late June 2017 the copyright registered distribution (there are three main FOAM distributions: FOAM-extend by H. Jasak, OpenFOAM from the OpenFOAM foundation by H. Weller, and OpenFOAM® by ESI group) of OpenFOAM® has made available an 'out of the box' working implementation of the overset technique in version 1706. These recent developments have had a positive impact for this work as they serve as a convenient benchmark for the overset One-way FSI implementation of this thesis. Specifically, three cases will be presented here and compared to the overset results obtained with OpenFOAM® version 1706: a piston-cylinder assembly, two oscillating cylinders enclosed in a rectangular cavity, and flow induced by two plates with general planar motion.

An effort has been made to set up the cases as similar as possible for both code implementations. This includes using alike grid resolutions, and the same discretization schemes for the time, advection, and diffusion terms in the governing equations. However, key differences are unavoidable for the hole cutting algorithm and the inter-grid communication procedure in the overlapping zone. The scarce theoretical documentation available for OpenFOAM® , along with the lack of time to delve into the details of the implementation, means that the standard setting has been used for the inter-grid interpolation cases using OpenFOAM® v 1706. This setting is called 'inverse distance' and it apparently is equivalent to linear interpolation in which the weights of the donor cells are computed using the inverse distance from donor to acceptor, whereas MFBI has been used in the overlapping zone in the current work. For the rest of this Chapter results obtained with the overset OpenFOAM® software will be referred to simply as 'OpenFOAM® ', and the results obtained with the in-house software shall be named 'OVERSET'. Lastly, all cases presented here are two-dimensional due to processing restrictions.

5.2 Piston-Cylinder Assembly

The first case is the analysis of the intake stroke of a piston-cylinder assembly presented in [87]. Although experimental data is presented in the original research paper, there appears to be an inconsistency in the reported Reynolds number defining the operating condition of the experiment. Because of this it has been decided to employ the OpenFOAM® solution as a benchmark. The geometry of the assembly and the coarse computational grid are shown in Figs. 5.1, 5.2. Referring to the latter, there are three grids employed for the OVERSET code case set up: the first one is associated with the inlet channel (blue), the second one covers the whole piston cavity (grey), and the third one discretizes the moving piston wall and extends back into the flow domain (red) where it will take precedence over the background piston cavity grid whenever possible. The total number of cells and time step employed for the three grid resolutions are: 2536 cells with $\Delta t = 5 \times 10^{-03}$ s (grid 0), 10144 cells with $\Delta t = 2 \times 10^{-03}$ s (grid 1), and 22824 cells with $\Delta t = 1 \times 10^{-03}$ s (grid 2). In all instances the advection based Courant number is less than one.

Dirichlet boundary conditions for the momentum equations are used on the leftmost vertical surface to represent an inlet with a uniform horizontal velocity profile, which satisfies global continuity based on the piston velocity. The top surfaces around the enclosure are set to zero Dirichlet values for the momentum equations to represent static walls, and the vertical piston surface is a wall moving with a constant velocity of 11.90 $\frac{mm}{s}$ in the positive horizontal direction. The bottom surface is set to symmetry which entails zero normal gradients of the velocity field, and the additional constraint of zero normal velocity as well. The inlet stroke takes place in 2.04 s and the flow is in laminar regime with $Re = 98$ based on the piston velocity and the vertical length of its cavity.

The OpenFOAM® case was set up to model the whole domain without assuming axial symmetry of the flow field, with the remaining boundary conditions being the same, and a similar grid resolution to the fine grid employed in the OVERSET code. The second order backward scheme was used for time discretization, the QUICK scheme for advection, and central differences for the diffusion terms.

The most interesting features of the flow field are shown in Fig. 5.3 through a sequence

of images depicting the evolution of the velocity magnitude field as the piston moves. The boundary layer in the small channel causes the centerline horizontal velocity component to increase until just after the area expansion section, where there is a pressure increase (this can be confirmed via Fig. 5.12) which causes the axial velocity component to drastically decrease. At the same time, the sudden area expansion allows the vertical velocity component to increase swiftly close to its corners, since the fluid is no longer bounded by the inlet channel walls. This, combined with the no slip condition along the vertical walls induces the formation of vortices in those areas as can be seen in the top right image of the sequence. These vortices are finally advected through the piston cavity as seen in the middle and bottom images of the figure.

For reference Fig. 5.4 presents the selected locations for the remaining profile plots of this case. Figures 5.5, 5.6, and 5.7 reveal there is practically no difference for the velocity profiles between the mid and high resolution grids, whereas assessment of grid sensitivity via the pressure field is not instructive for this particular case. This is because the reference pressure location was set to the cell with indices $i = 2$ $j = 2$, and $k = 2$ which effectively has different coordinates for every grid refinement. Still the plot is presented for illustration purposes. These plots are profiles along a vertical line located 4 *mm* after the channel expansion 650 *ms* after the moving piston has started its course. For the following comparisons with the OpenFOAM® code the high grid resolution shall be employed exclusively.

A comparison of the velocity magnitude field between the two codes via Figures 5.8, and 5.9 at a simulation time of 0.510 *s* reveals both solutions to be in close agreement. The pressure fields (Figs. 5.10, 5.11) at the end of the simulation ($t = 2.04$ *s*) also exhibit good agreement, with the differences in magnitude due to different locations of the cell selected to provide a reference value for pressure . This can be most notably appreciated by examining Figures 5.12 to 5.14 which present the pressure profiles for both solutions at different times and locations throughout the simulation course. The patterns are exactly the same with their magnitudes slightly shifted.

Axial and vertical velocity profiles are shown in Figures 5.15 through 5.18 and confirm both solutions are in agreement. This is encouraging and, indeed, the level of agreement

is remarkable considering the potential differences between both codes regarding inter-grid interpolation, and the definition of active, interpolation, and inactive cells through the hole cutting algorithm.

Due to the nature of the simple linear constant-velocity piston motion, the grid deformation approach discussed in Chapter 2 could also be a feasible methodology to solve this problem. However, its use would require cautious consideration regarding the piston-expansion amplitude range. Even though piston wall-adjacent cells would be impervious to orthogonal degradation imposed by grid deformation, the expanding motion would inevitably lead to cell coarsening in the piston wall-normal direction. If the range of motion is excessive this would significantly impact the accuracy of the predicted flow field since the most pronounced gradients are in such a direction owing to piston movement. On the other hand, the tradeoff would most likely be in terms of computation time as the costly hole-cutting algorithm required for overset grids would not need to be invoked.

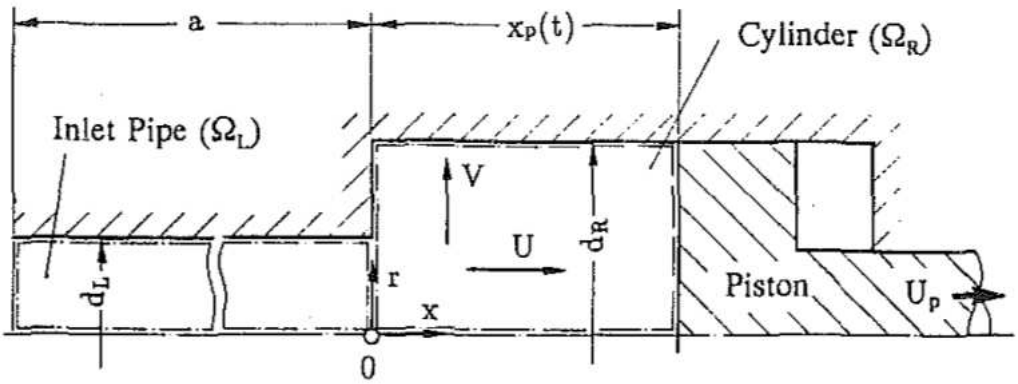

Figure 5.1: Geometry of the problem. $d_R = 45\ mm$, $d_L = 19\ mm$, $a = 100\ mm$, $x_{po} = 40\ mm$, $x_{pmax} = 64.3\ mm$. Image from [87].

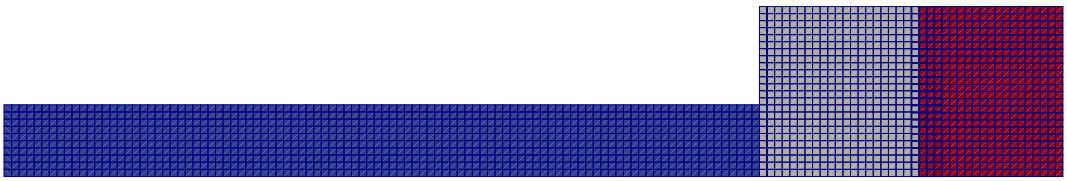

Figure 5.2: Coarse grids for the Overset code simulation

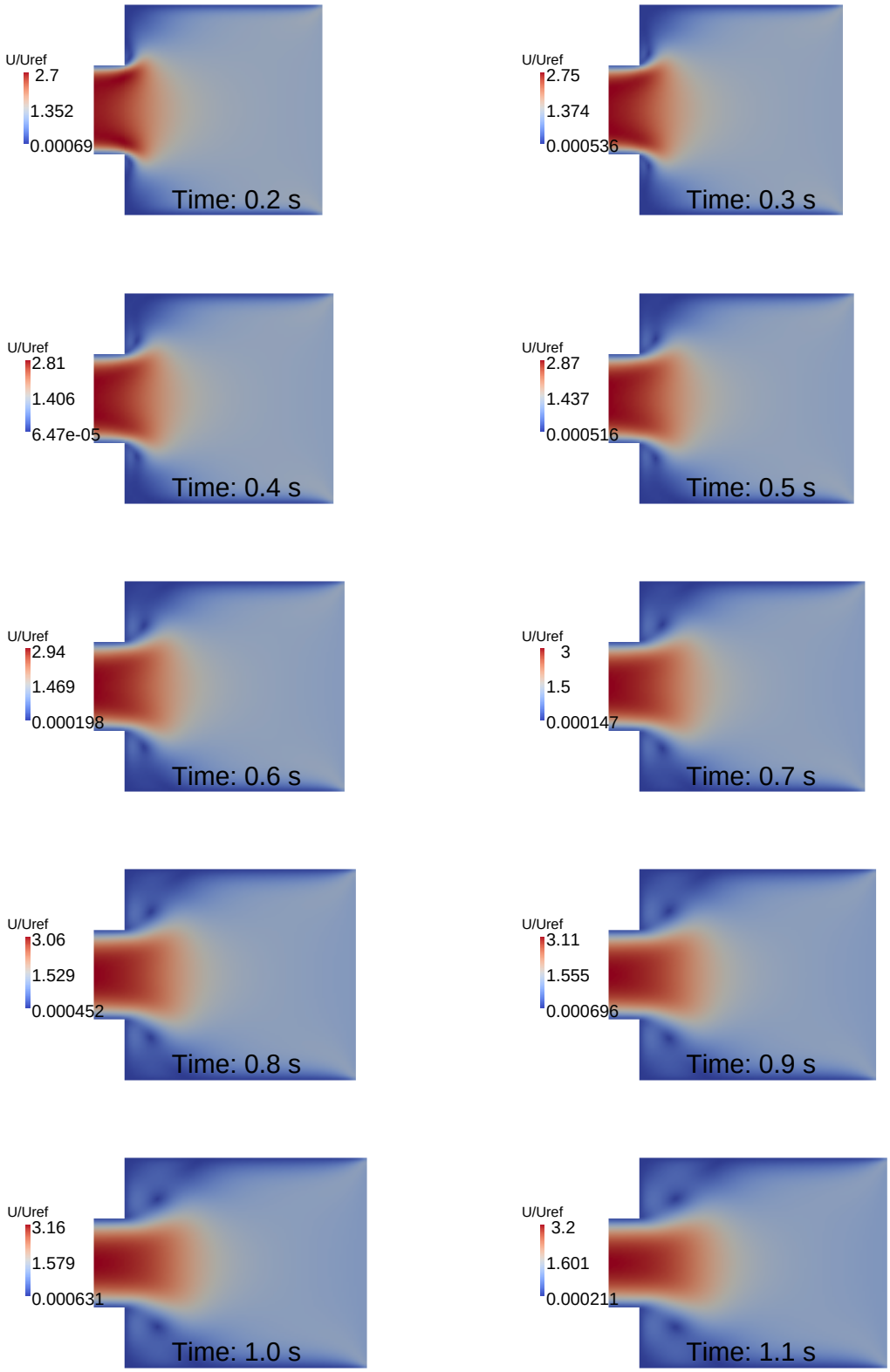

Figure 5.3: Evolution of the velocity magnitude field . Left to right, top to bottom. From $t = 0.20\ s$ to $t = 1.10\ s$.

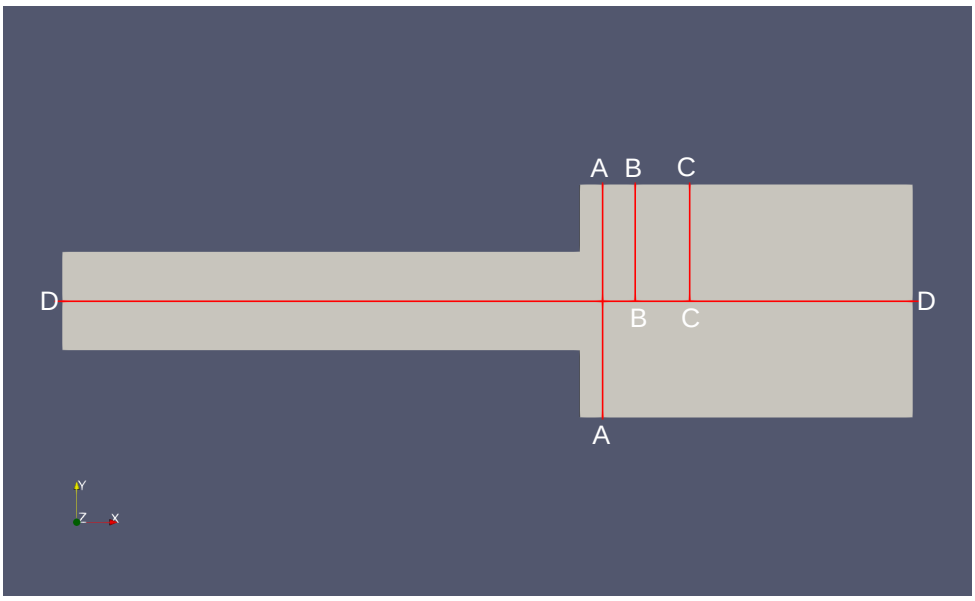

Figure 5.4: Locations of the ploted profiles: A-A $x = 4\ mm$ $(x/D = 0.088)$, B-B $x = 1.0170\ cm$, C-C $x = 2.0470\ cm$, and D-D $y = 0\ cm$.

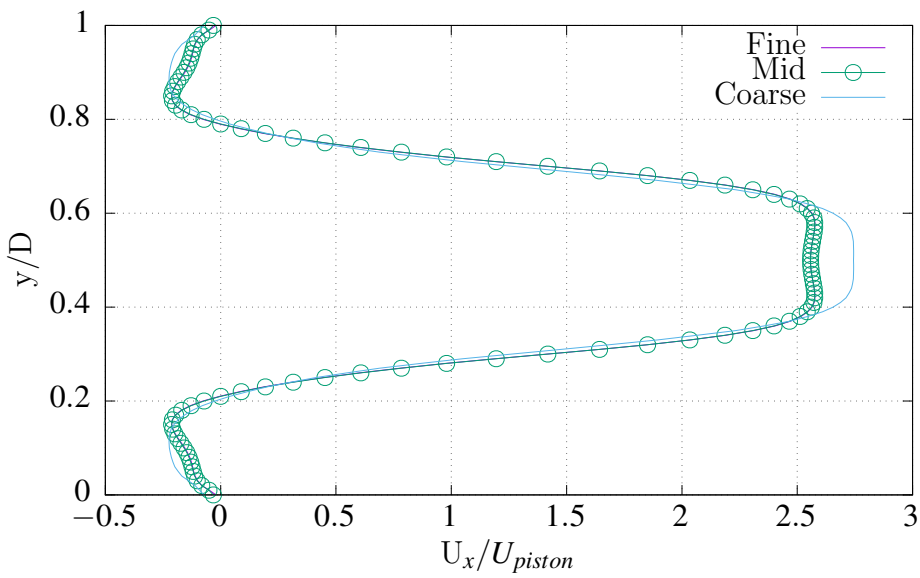

Figure 5.5: Grid sensitivity for horizontal velocity component at $\frac{x}{D} = 0.088$, $t = 650\ ms$.

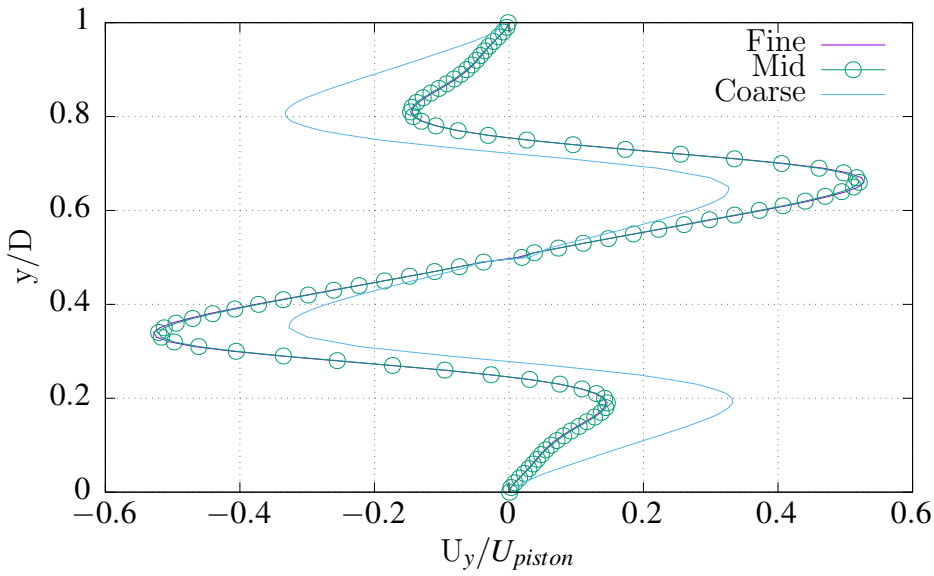

Figure 5.6: Grid sensitivity for vertical velocity component at $\frac{x}{D} = 0.088$, $t = 650\ ms$.

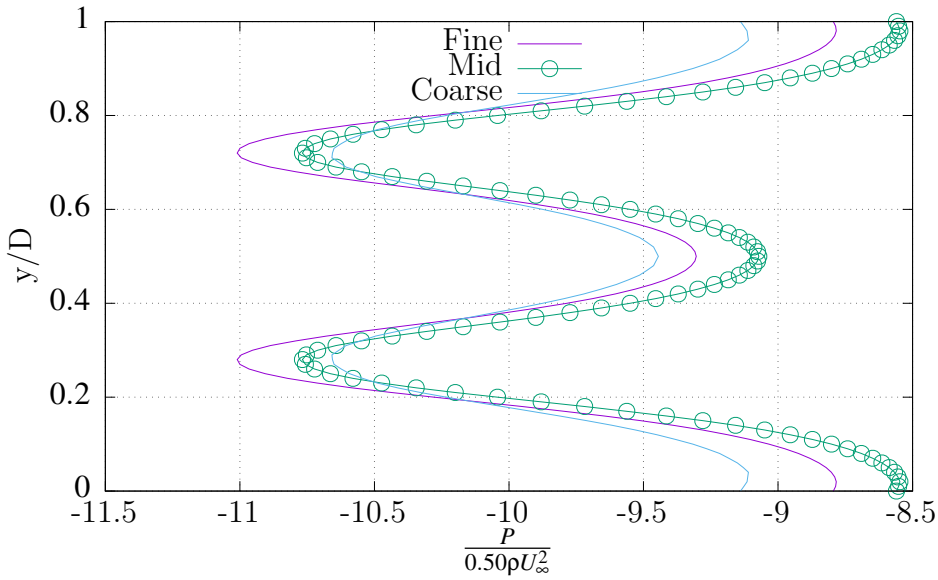

Figure 5.7: Grid sensitivity for pressure at $\frac{x}{D} = 0.088$, $t = 650\ ms$.

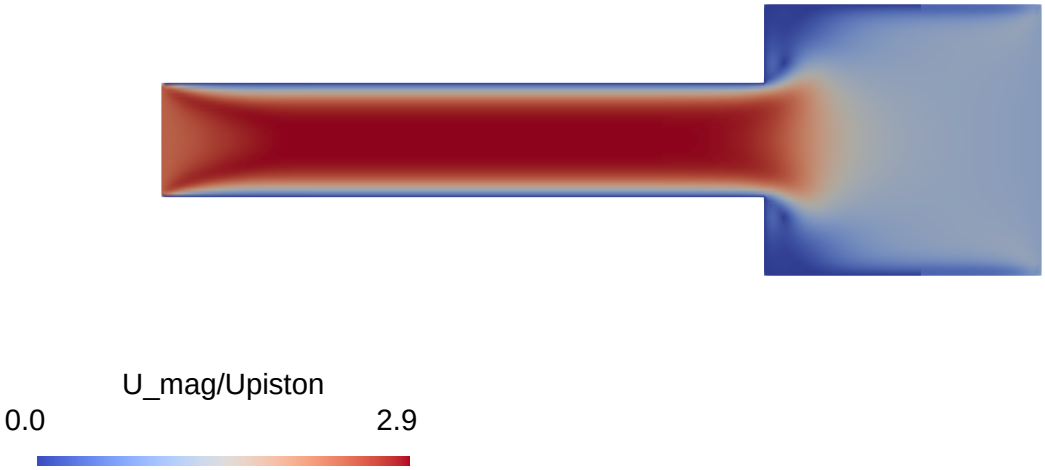

Figure 5.8: OpenFOAM® velocity magnitude at time 0.510 s.

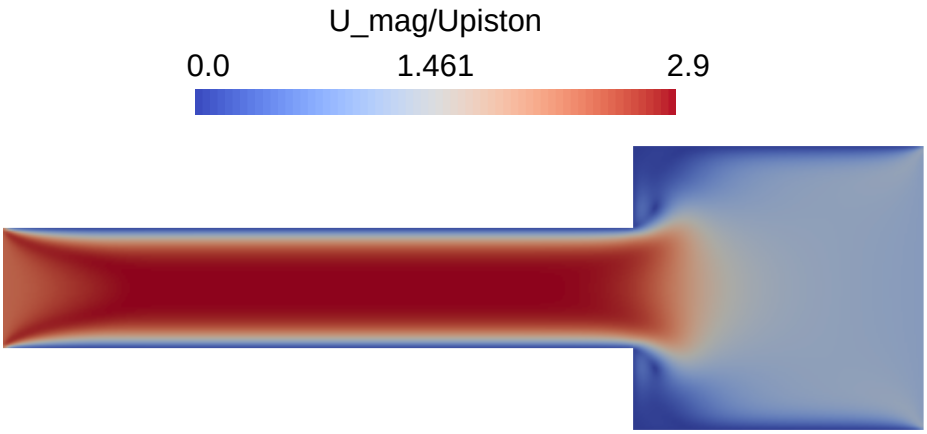

Figure 5.9: Overset code velocity magnitude at time 0.510s.

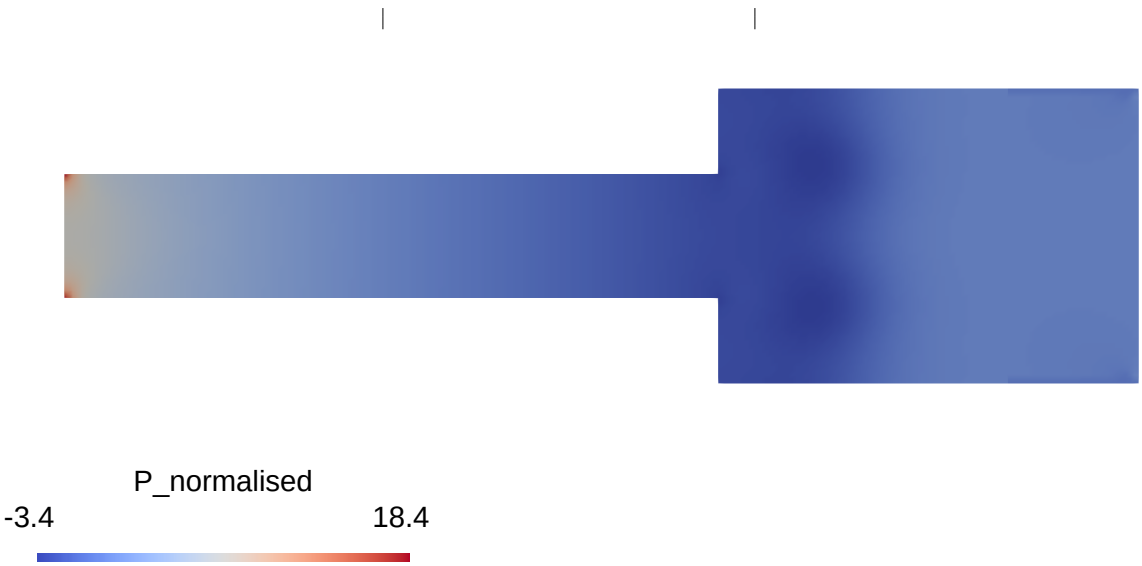

Figure 5.10: OpenFOAM® pressure field time 2.04 s.

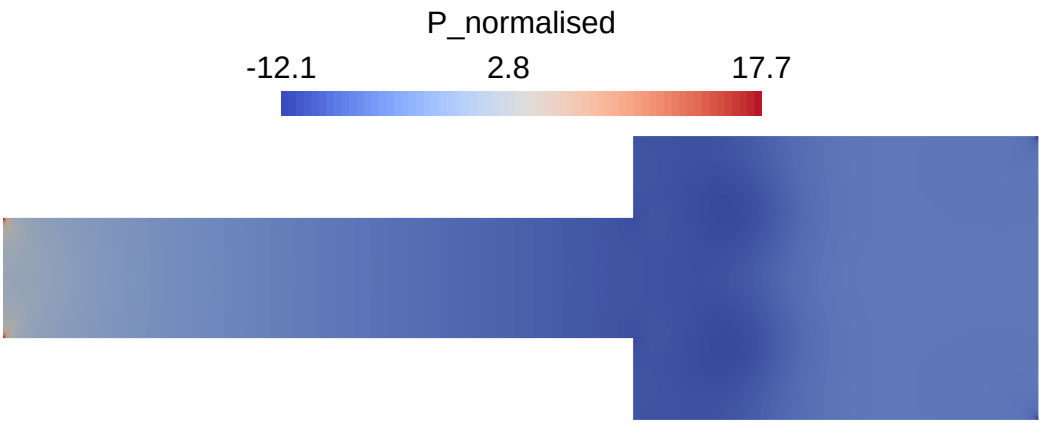

Figure 5.11: Overset pressure field time 2.04 s.

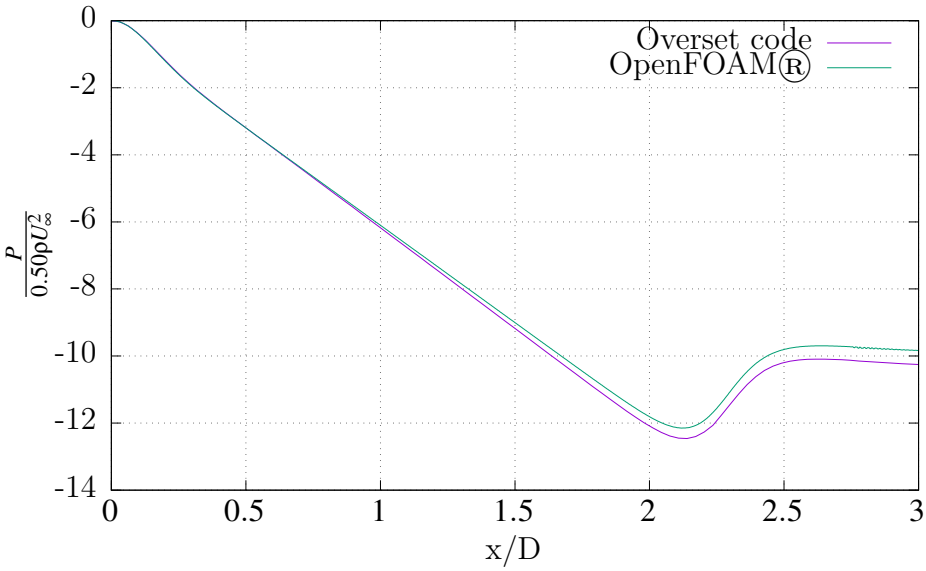

Figure 5.12: Pressure profile comparison at time 0.310 s, $y = 0$.

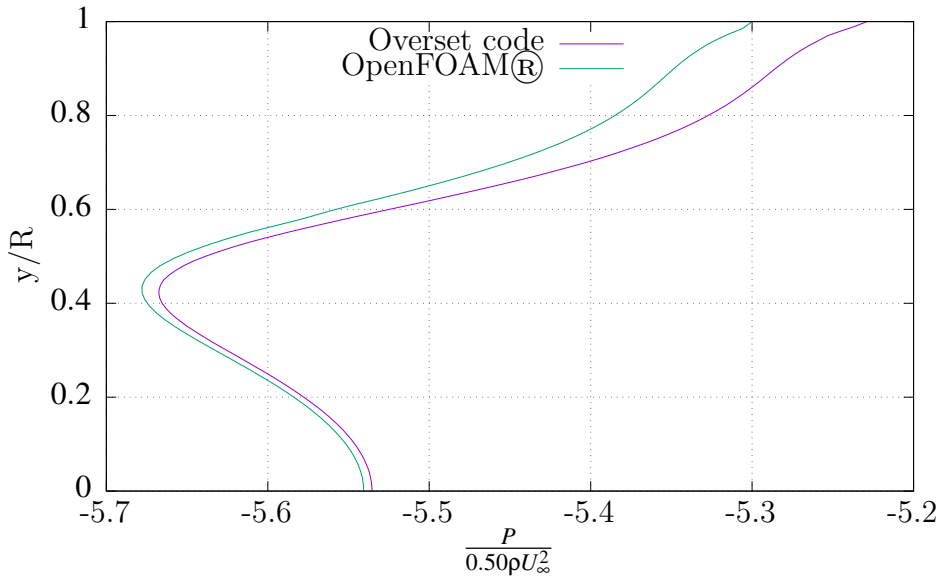

Figure 5.13: Pressure profile comparison at time 0.510 s, $x = 1.0170$ cm.

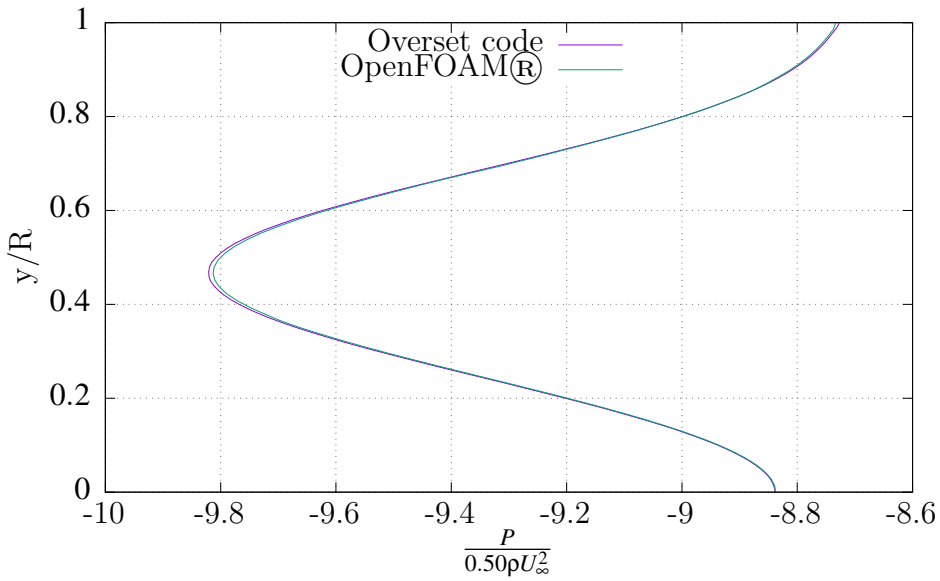

Figure 5.14: Pressure profile comparison at time 2.04 s, $x = 2.0470$ cm.

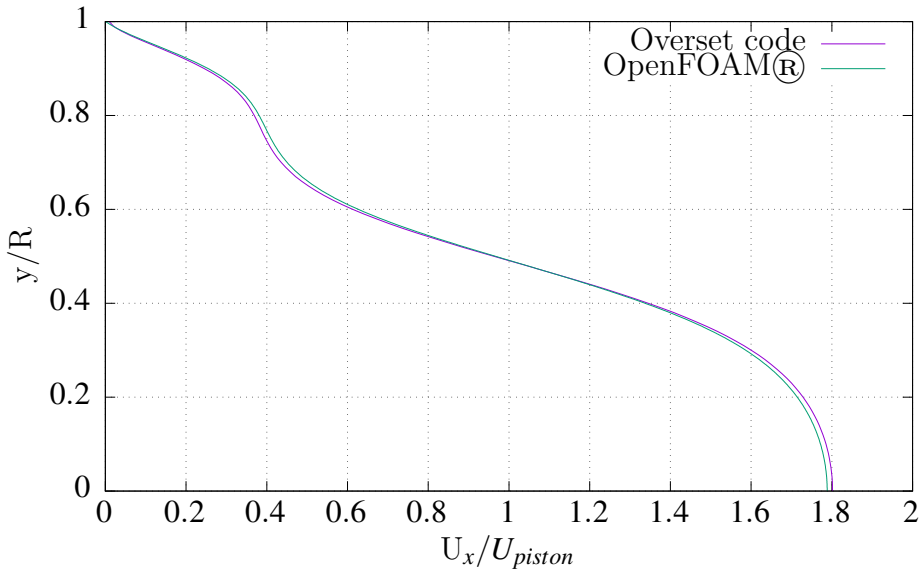

Figure 5.15: Horizontal velocity profile comparison, time 0.510 s, $x = 1.0170$ cm

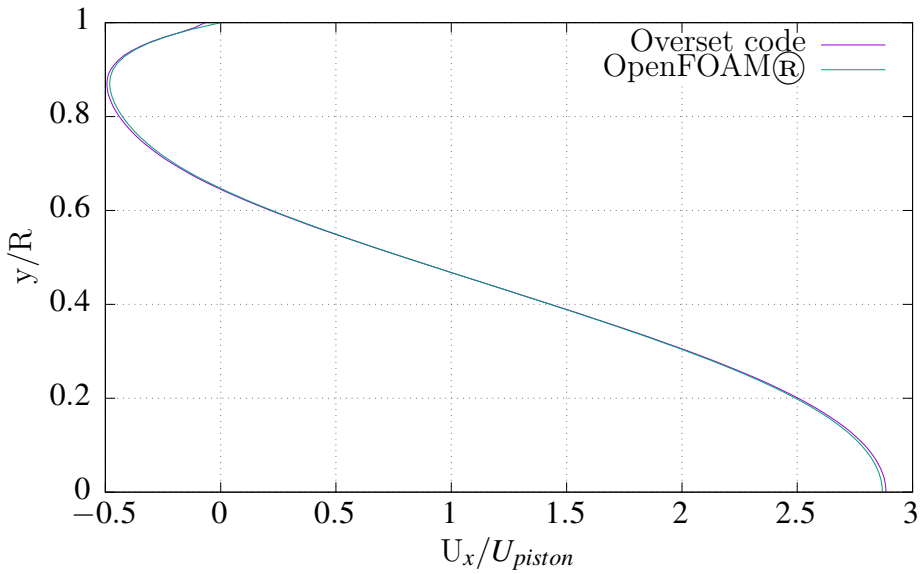

Figure 5.16: Horizontal velocity profile comparison, time 2.04 s, $x = 2.0470$ cm

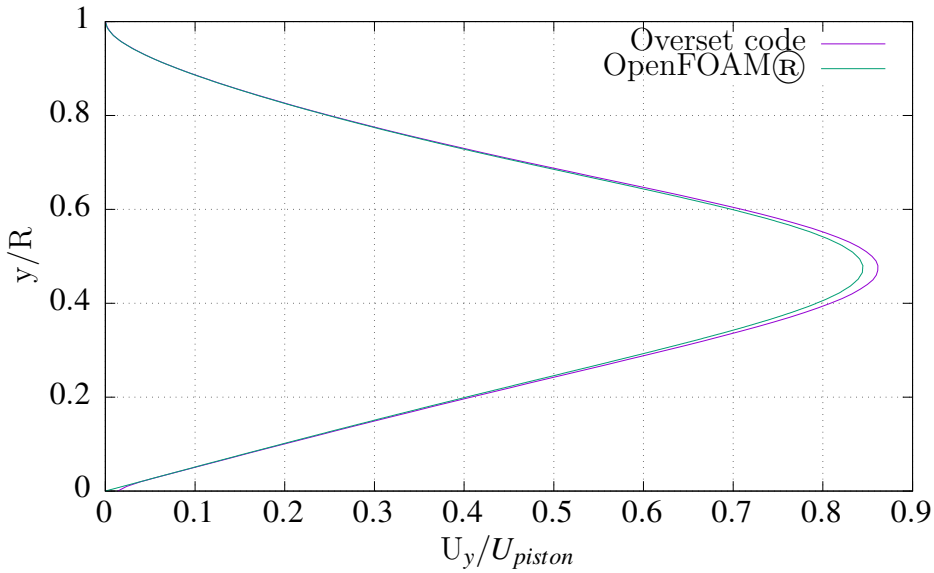

Figure 5.17: Vertical velocity profile comparison, time 0.510 s, $x = 1.0170$ cm

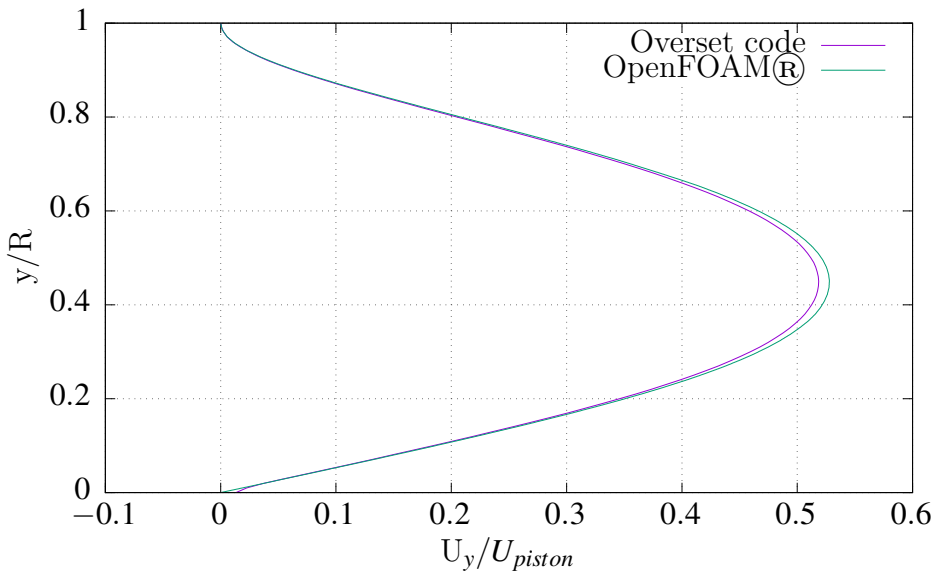

Figure 5.18: Vertical velocity profile comparison, time 2.04 s, $x = 2.0470$ cm

5.3 Oscillating Cylinders

Consider the two dimensional flow induced by two moving cylinders enclosed in a rectangular cavity. The cylinders, initially separated horizontally by 16 m are set in a prescribed oscillatory motion about their initial positions for 16 s. After this, they travel in the horizontal direction towards each other with constant velocity, and eventually end up separated by the same initial distance but in opposite positions after 32 s. This case was studied using the immersed boundary method in [88], and serves as a challenging test for the hole cutting algorithm when the cylinders are close to each other because three grids are effectively overlapping at times. Figure 5.19 shows the geometrical parameters of the problem, while Fig. 5.20 shows an outline of the coarse computational grids. As seen in the figure, there is a background mesh to discretize the rectangular cavity, and a body fitted grid for each cylinder which takes precedence over the background grid at the near cylinder regions. Two grid resolutions were explored for this simulation: 15552 cells with $\Delta t = 1.25 \times 10^{-02}$ s, and 32040 cells with $\Delta t = 1.25 \times 10^{-02}$ s. The high resolution grid however is not a consequence of a direct refinement of the coarse grid for this particular problem, the differences amongst the two grids are largely found in the background mesh, with the body fitted cylinder grids remaining almost the same in both cases. The main objective of doing this was to try to prevent as much as possible the appearance of cells in the current time computation that were inactive in one or two previous time steps, to avoid having to introduce additional interpolation. This was particularly troublesome when both cylinders were close to each other as seen through Figs. 5.21 and 5.22, where outlines of the holes created by Skillen's algorithm when the cylinders are close to each other are presented. However, it must be mentioned that in such cases 'previous' or 'previous previous' interpolation stencils were always found correctly to provide the required scalar values. Since the body fitted grids remained basically the same in both cases, the Courant number was still below 1 for the fine grid even when using the same time step.

The prescribed motion for the lower and upper cylinders is given respectively by:

$$x_{lc} = \begin{cases} \frac{4}{\pi}\sin(\frac{\pi t}{4}) & 0 \leq t \leq 16, \\ t - 16 & 16 \leq t \leq 32, \end{cases}$$

$$x_{uc} = \begin{cases} 16 - \frac{4}{\pi}\sin(\frac{\pi t}{4}) & 0 \leq t \leq 16, \\ 32 - t & 16 \leq t \leq 32 \end{cases}$$

Zero velocity Dirichlet boundary conditions are set for momentum equations on the static cavity walls, whereas time-dependent Dirichlet values are set on each cylinder surface according the prescribed velocity conditions. The Reynolds number based on the constant velocity of the cylinders once they are approaching each other, and their diameter, is $Re = 40$. The OpenFOAM® case was set up using exactly the same fine background mesh with around 30 cells more for each body fitted cylinder grid, and the same time step of 1.25×10^{-02} s was used. The backward time discretization scheme along with the QUICK convection discretization scheme was used in both codes.

During the first half of the simulation the flow is marked by the oscillating nature of the prescribed motion of each cylinder. Initially, as each cylinder accelerates towards its maximum velocity, the fluid in the vicinity of its surface attains considerable momentum in the same direction as the cylinder is traveling, forming a wake just behind its trailing surface. This can be observed in the first frame of Fig. 5.23. However, as soon as the cylinder starts to move in the opposite direction it splits this wake and thin regions of high velocity gradients are found just above and below the cylinder. This is observed in the second frame of the Figure. As the cylinder now reaches its maximum velocity in the current direction the remains of the original wake are further displaced as seen in the last frame of this Figure.

The highlight of the second half is the period during which both cylinders approach each other, illustrated in Fig. 5.24 via velocity magnitude contours. As both cylinders come closer their respective vertical forces act to create a 'repulsive' effect as revealed in Fig. 5.25. However, once they are sufficiently close a low pressure region develops in the middle gap, reversing the trend, and they 'attract' each other. It is worth clarifying the fact that this is a one-way simulation and the cylinder motion is restricted to the horizontal direction. That is the reason why an actual displacement in the vertical direction is not seen in the images.

Figure 5.26 is a comparison of the drag coefficient of the lower cylinder using both

coarse and fine grids. Both coefficients are equal and reflect the oscillating nature of the cylinder motion in the horizontal direction; when the cylinder advances in the positive axial direction the fluid exerts a negative force and vice versa. This result is somehow expected since the cell number for the cylinder body fitted grids in both cases differs only slightly. However, Figure 5.27 provides a clear picture of both grid resolutions, since it plots the cell centre values of the horizontal velocity component along a vertical line located at $x = 1m$ throughout the length of the cavity at $t = 12\ s$. As stated in the introduction of this Chapter, the goal of this set of problems is to benchmark the current overset implementation with OpenFOAM® , not to accurately resolve the physics via systematic grid refinement. In order to benchmark the overset code with OpenFOAM® the fine grid results will be used in the remainder of this section.

The selected locations to plot solution profiles for comparison purposes with OpenFOAM® are presented along with the velocity magnitude field in Figs. 5.28, and 5.29. Figures 5.30 to 5.32 are solution profiles along a vertical line covering the length of the cavity, located a short distance (0.20 m) behind the lower cylinder at a simulation time of 1 s. At this stage the lower cylinder is travelling in the positive x direction, the horizontal velocity component in Figure 5.30 reveals there are recirculation regions above and below the cylinder wake. However, these recirculation zones vanish and there is a rapid change in the central region of the wake which reflects the nature of the positive horizontal velocity of the cylinder. In Figure 5.31, two regions of high (opposite) vertical velocity component that were detached from the cylinder's surface are being advected outwardly off the main wake, which as expected has zero vertical velocity in its centre. Figures 5.33 through 5.35 are the equivalent profiles for the upper cylinder at time $t = 9\ s$ when it is travelling towards the negative x direction. Throughout this series of profiles both solutions exhibit excellent agreement. It is worth noting there are some non-physical pressure oscillations in Figures 5.32 and 5.35 for both codes, most notably in the OpenFOAM® solution. These oscillations emerge due to the coarseness of the employed grids in the overlapping zone, and the inter-grid interpolation algorithms which are most likely different for both codes. The overset code solution is using the MFBI which is why it presents oscillations to a lesser degree.

Figures 5.36 and 5.37 are useful to provide a more general panorama about how both solutions compare via the velocity magnitude field at $t = 15s$. Generally, as the cylinders move they encounter a high pressure zone in the direction they are heading towards. This is expected as the fluid in the vicinity of that region has low velocity, meanwhile the fluid in contact with the cylinder attains its velocity. High vorticity regions are found a short distance downstream along the front end of each cylinder surface. Nonetheless, these high vorticity zones rapidly vanish further downstream as the high diffusive momentum flux prevents the existence of regions with high velocity gradients. Qualitatively both solutions exhibit the same flow features whilst quantitatively there is a difference of about 5% in the peak magnitude value.

A comparison of the the upper cylinder lift and drag coefficients as both bodies approach, with the original solution obtained using the Immersed Boundary method in [88] is presented in Figures 5.38 and 5.39 respectively. As both solid objects move closer to each other there is a steady increase in the upper cylinder's lift coefficient until just after the lower cylinder is directly below. At this point there is a pronounced low pressure region in the vicinity of its bottom surface which causes both the lift and drag forces to decrease sharply. The two solutions are in close agreement for the lift coefficient while there is more difference regarding the drag coefficient. The Immersed Boundary solution was computed on a much finer computational mesh ($\approx$ 200000 cells). This case represents a far more challenging test to the hole cutting algorithm than the earlier moving piston one, especially during the approaching stage. Nevertheless, the comparison with both OpenFOAM® , and the Immersed Boundary method, reveals the current overset implementation is capable of tackling one-way multi-body simulations.

Finally, it is worthwhile to mention the main advantages/drawbacks of both the overset and immersed boundary approaches as seen through this case. The main drawback of the reference immerse boundary solution presented in [88] is the appearance of a flow field solution inside the solid-body structure. This unphysical feature is typical of immersed boundary techniques, and it is a direct consequence of the disperse fluid-solid interface resolution of the methodology which can also introduce errors leading to incorrect lift

and drag forces experienced by the cylinders when coarse grids are employed. This in exchange for high-quality and easily-generated cartesian grids throughout the flow domain. On the other hand, the overset-grid method solution can exhibit discontinuities and unphysical pressure oscillations in the overlapping region if coarse-grid sizing and standard interpolation inter-grid communication algorithms are employed respectively. However, in this case these undesirable characteristics appear far from the high-gradient regions located in the vicinity of the moving-solid boundaries, which are resolved accurately using boundary conforming structured high-quality grids with the overset method.

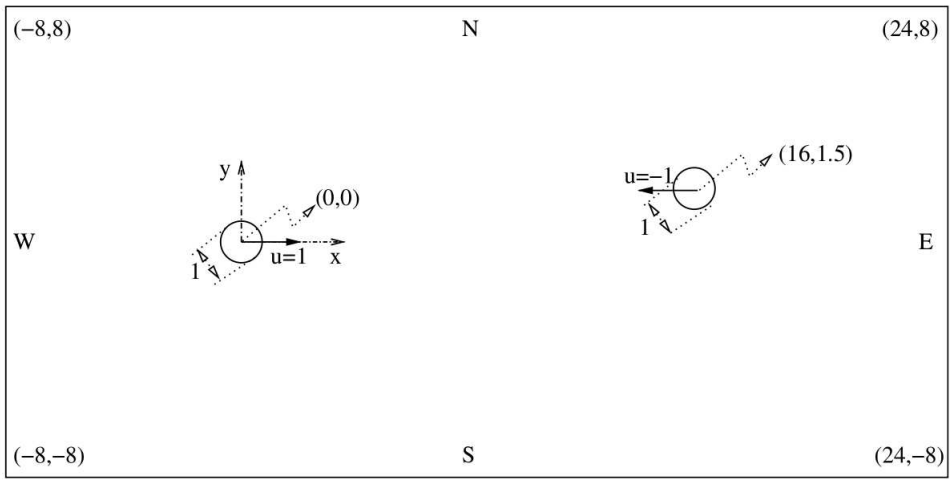

Figure 5.19: Geometry for the oscillating cylinders problem. Image from [88]

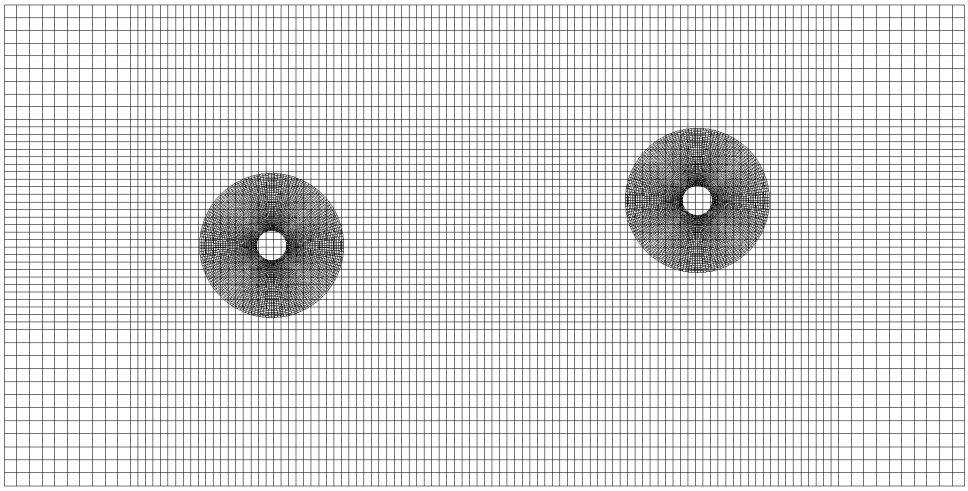

Figure 5.20: Coarse computational domain for the Overset code.

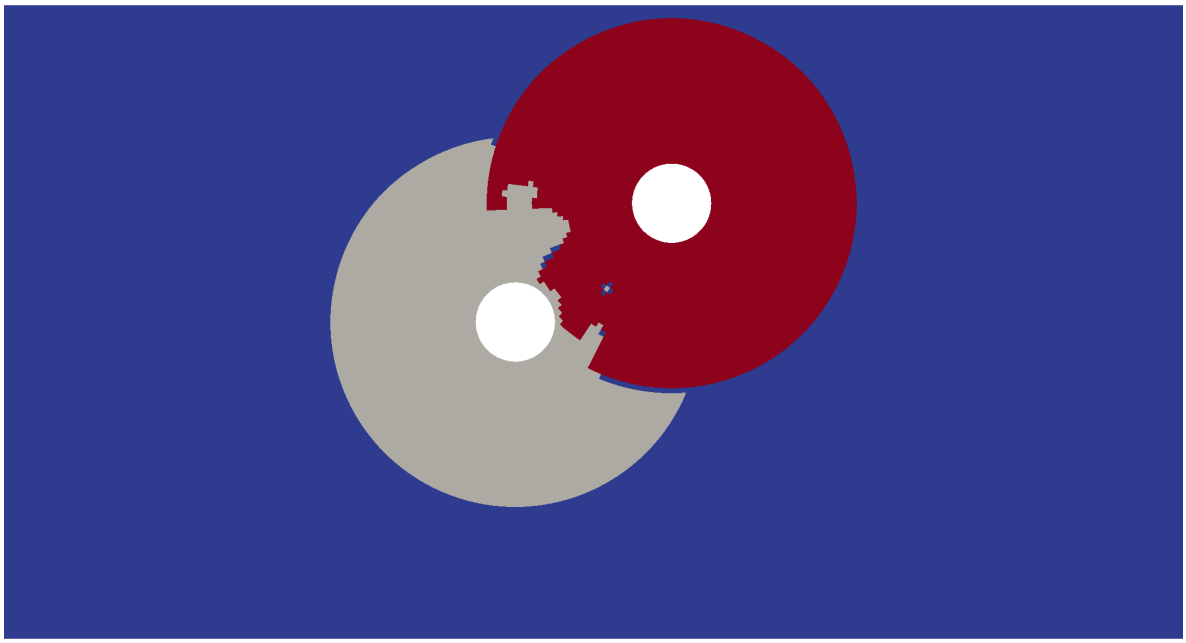

Figure 5.21: Overlapping zones at $t = 23\ s$.

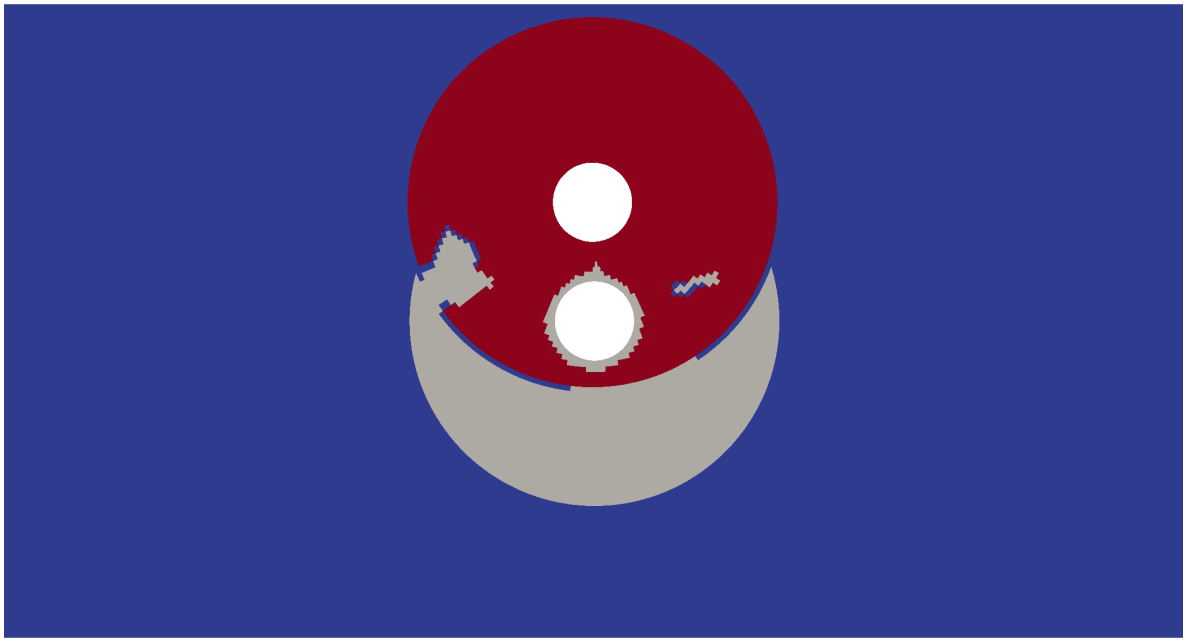

Figure 5.22: Overlapping zones at $t = 24\ s$.

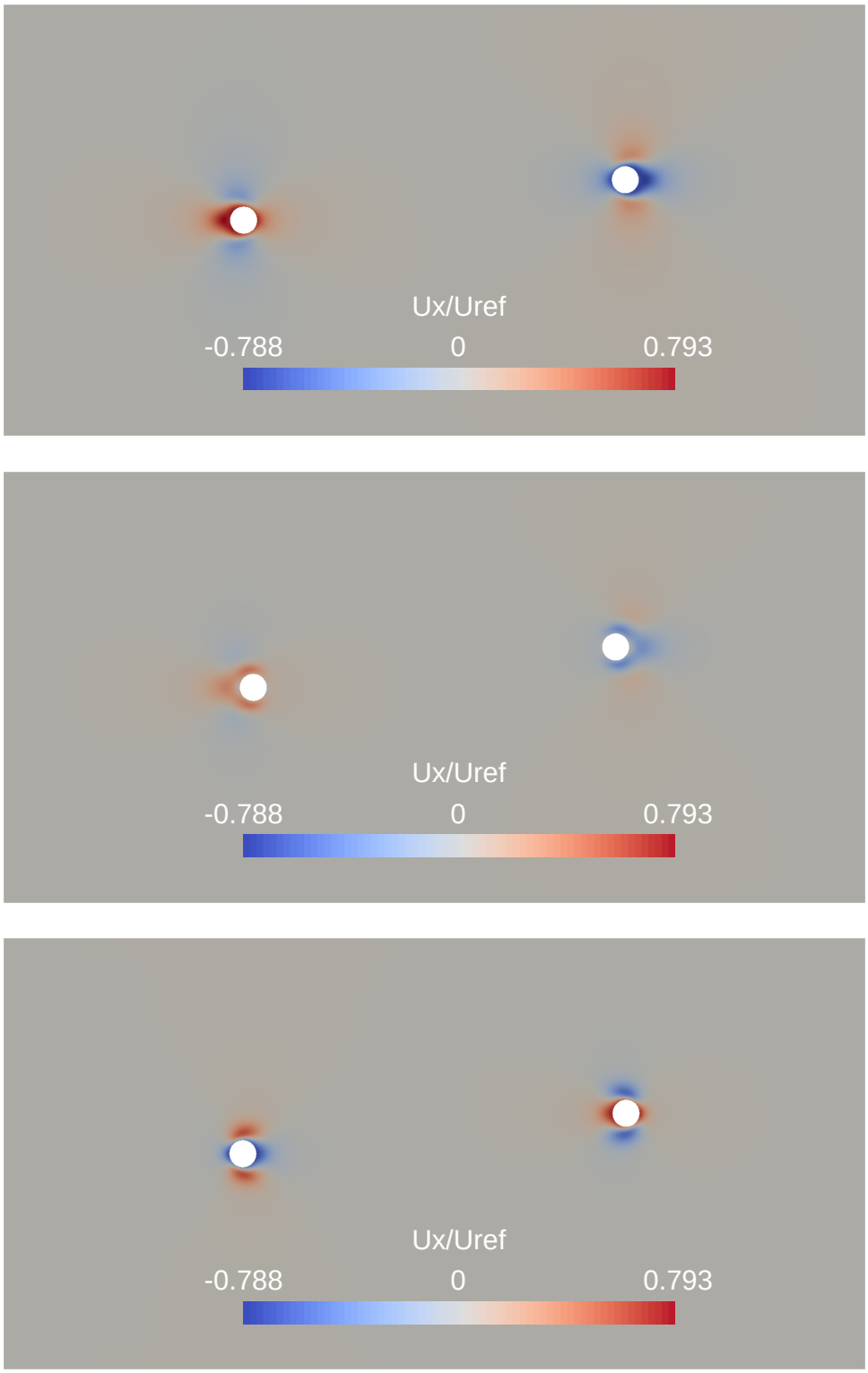

Figure 5.23: Features of the horizontal velocity component of the flow ($t = 1\ s$ top, $t = 2\ s$ middle, $t = 3\ s$ bottom)

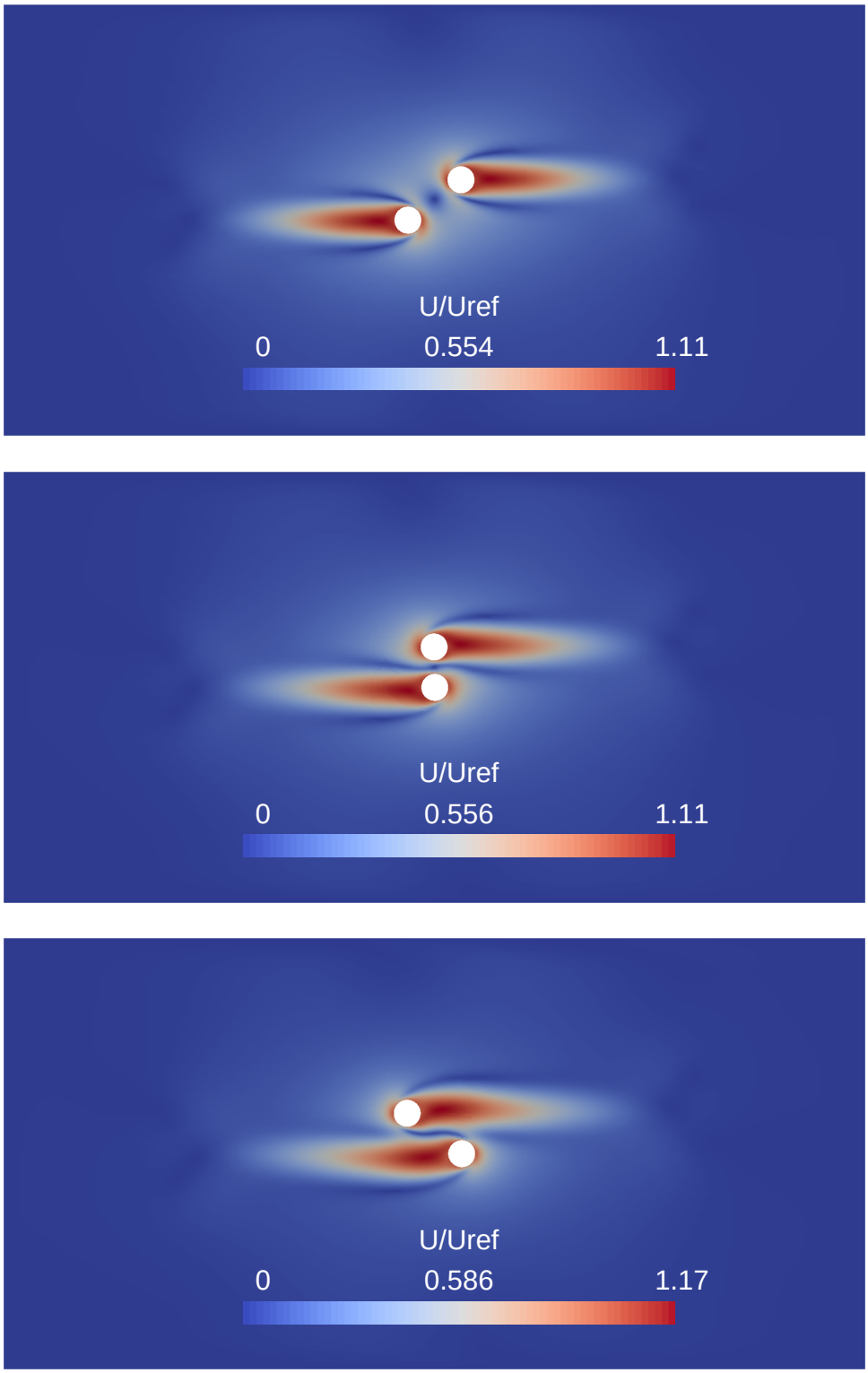

Figure 5.24: Evolution of the velocity magnitude field ($t = 23\ s$ top, $t = 24\ s$ middle, $t = 25\ s$ bottom)

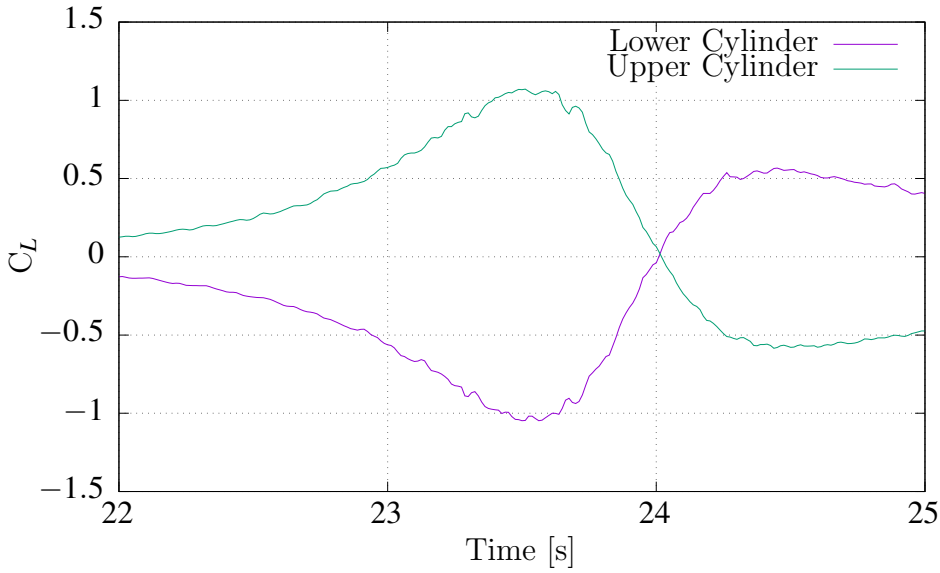

Figure 5.25: Lift coefficient of both cylinders during their approach.

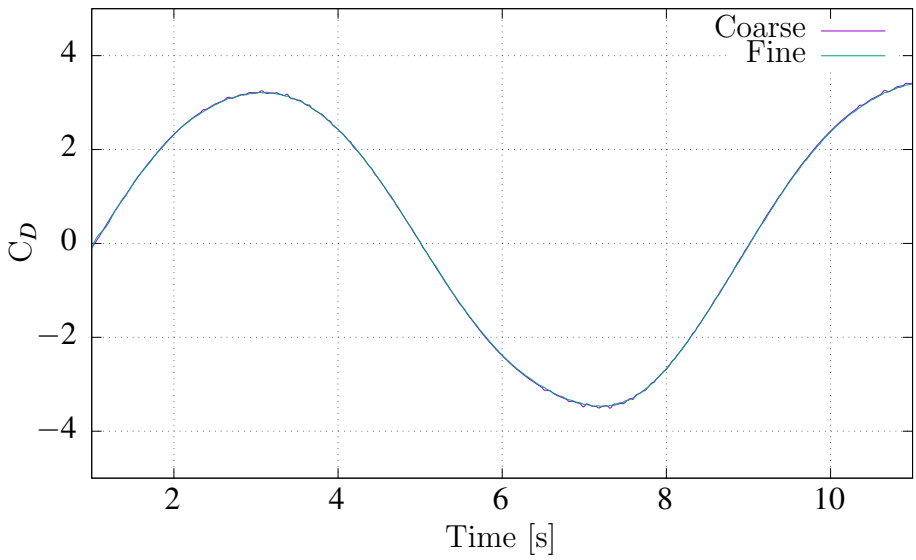

Figure 5.26: Drag coefficient of the lower cylinder using coarse and fine grids for the Overset code.

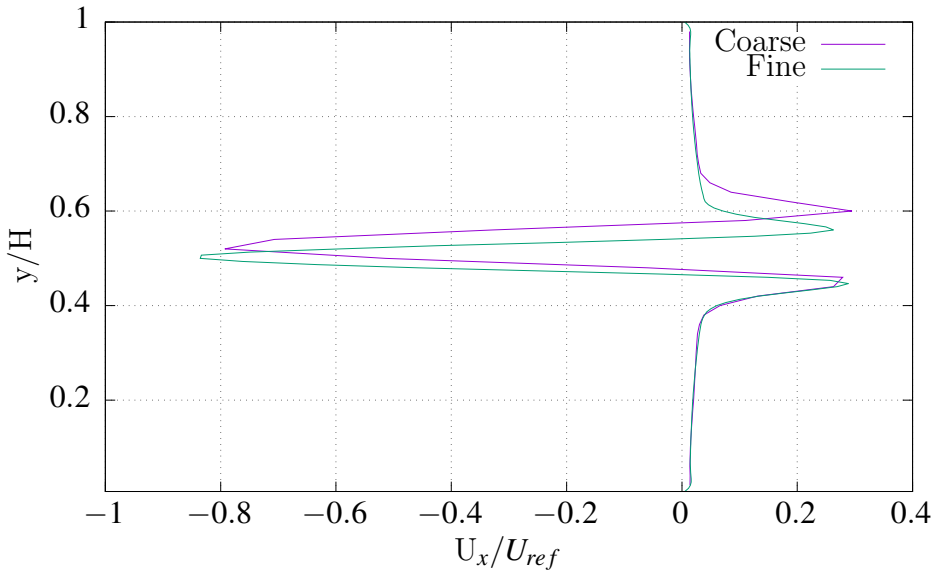

Figure 5.27: Fine and coarse horizontal velocity profiles (cell centre values) at $t = 12\ s$.

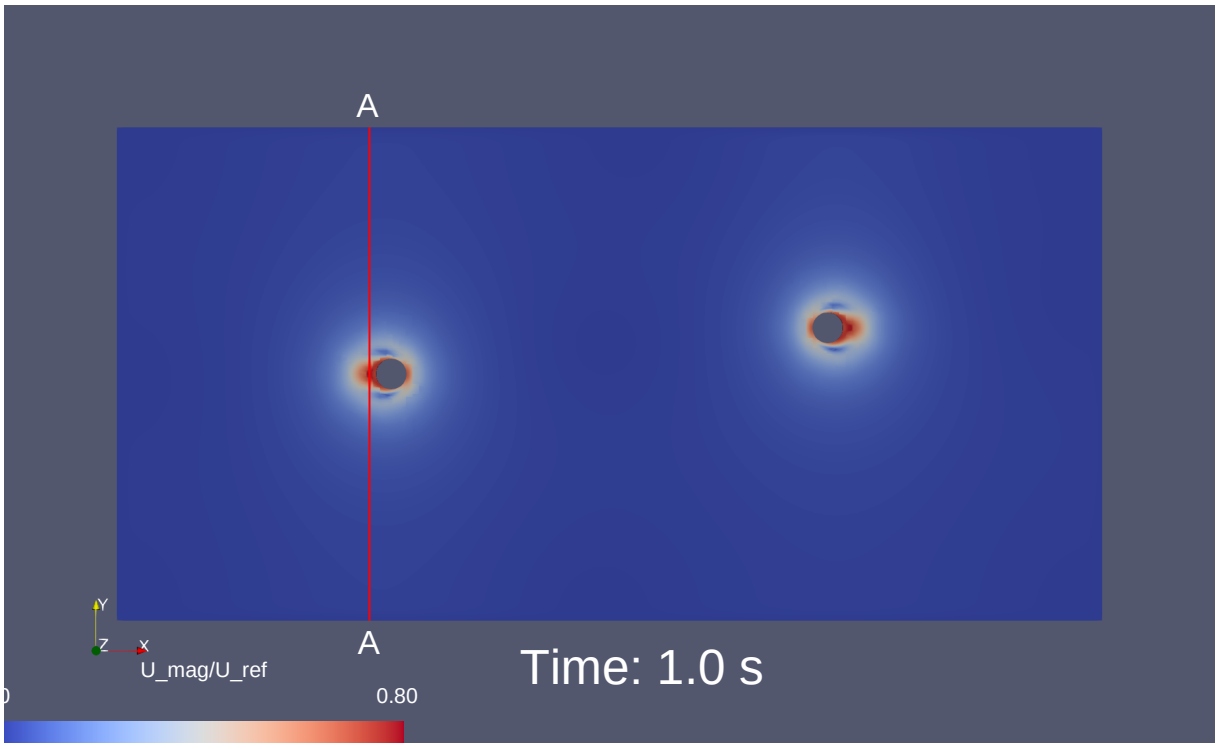

Figure 5.28: Velocity magnitude and selected location (line A-A) to plot solution profiles at: $x = 0.2\ m$, $t = 1\ s$.

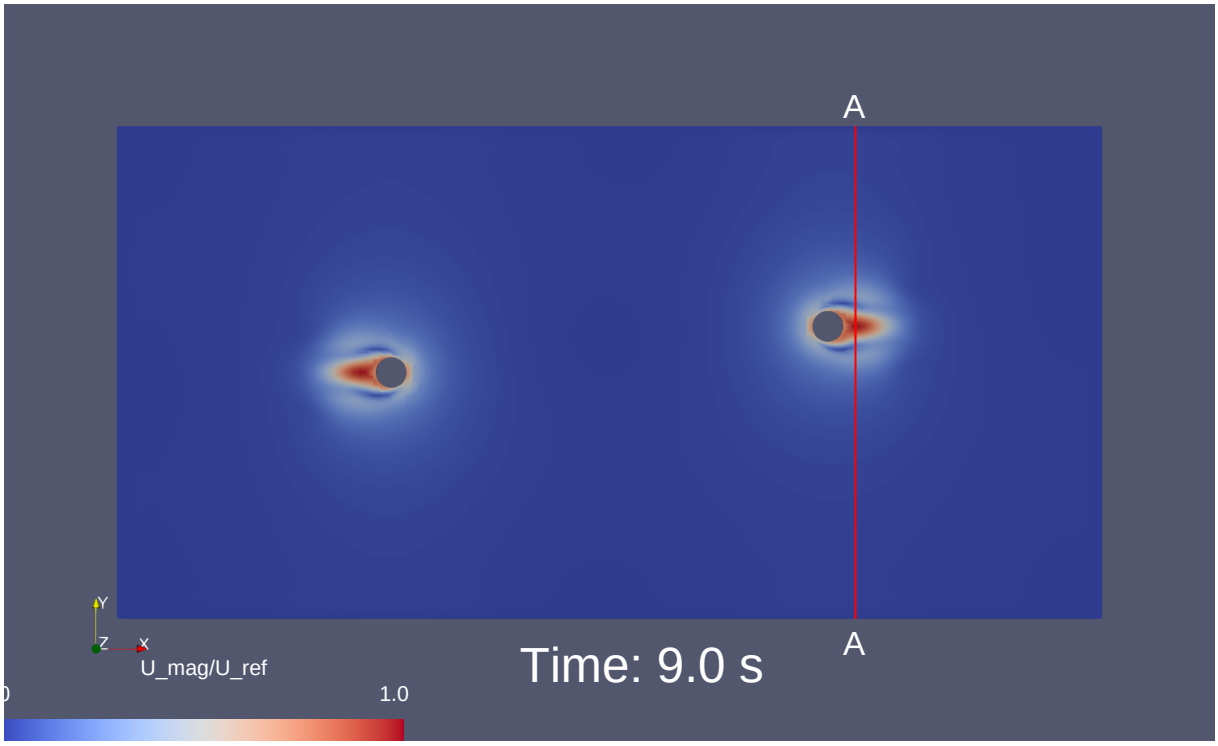

Figure 5.29: Velocity magnitude and selected location (line A-A) to plot solution profiles at: $x = 16\ m$, $t = 9\ s$.

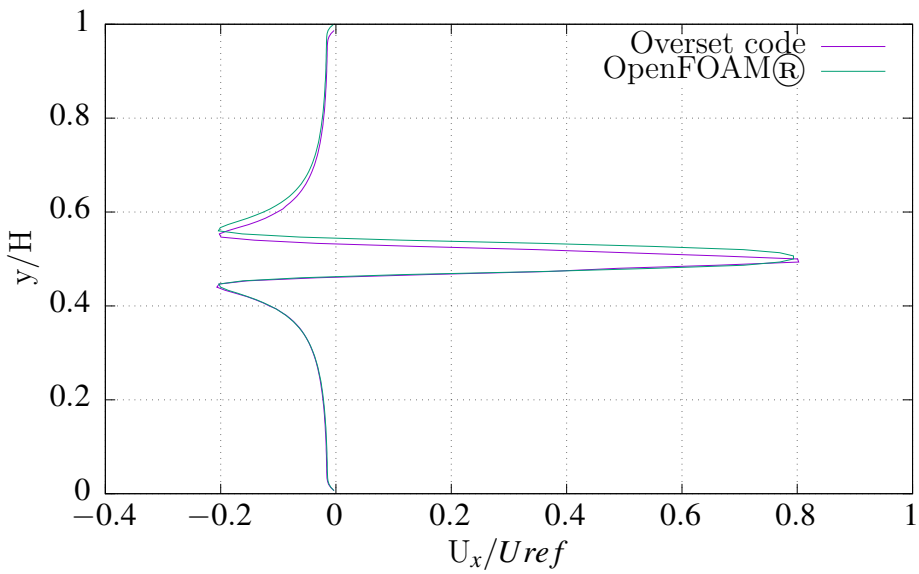

Figure 5.30: Horizontal velocity profiles. $x = 0.2\ m$, $t = 1\ s$.

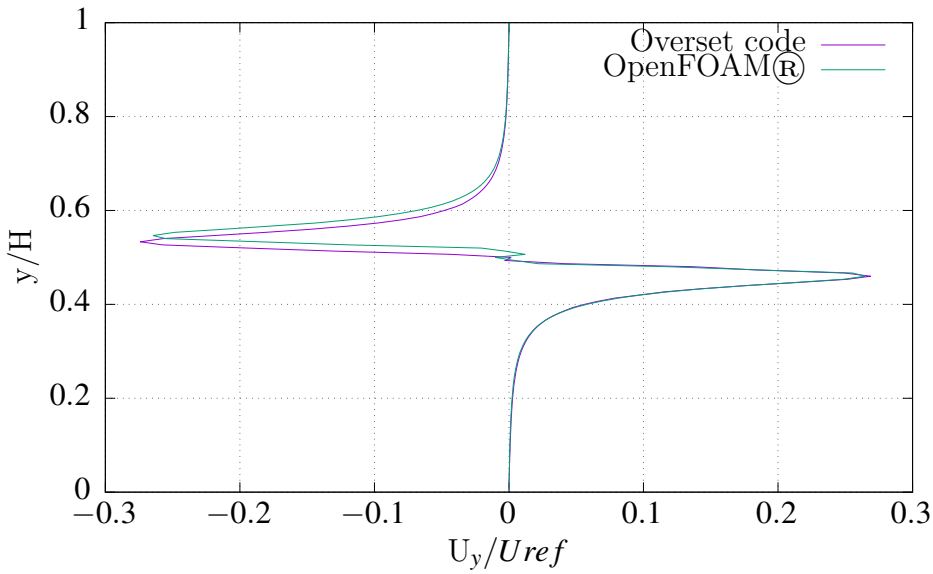

Figure 5.31: Vertical velocity profiles. $x = 0.2\ m$, $t = 1\ s$.

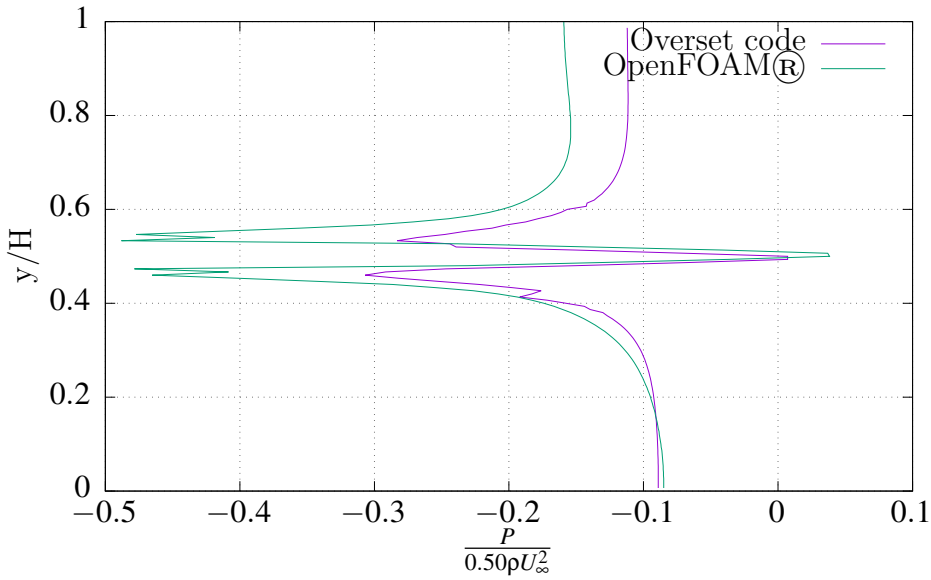

Figure 5.32: Pressure profiles. $x = 0.2\ m$, $t = 1\ s$.

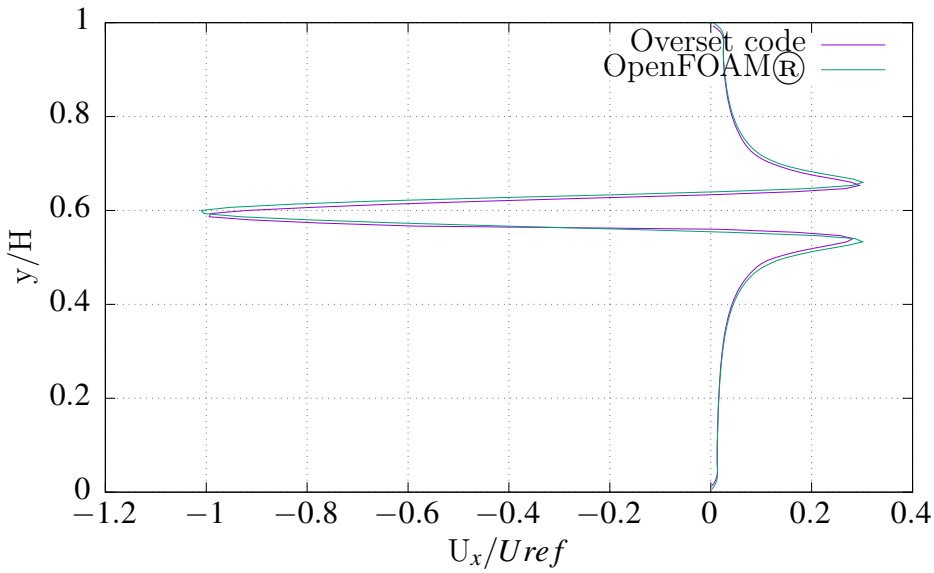

Figure 5.33: Horizontal velocity profiles. $x = 16\ m$, $t = 9\ s$.

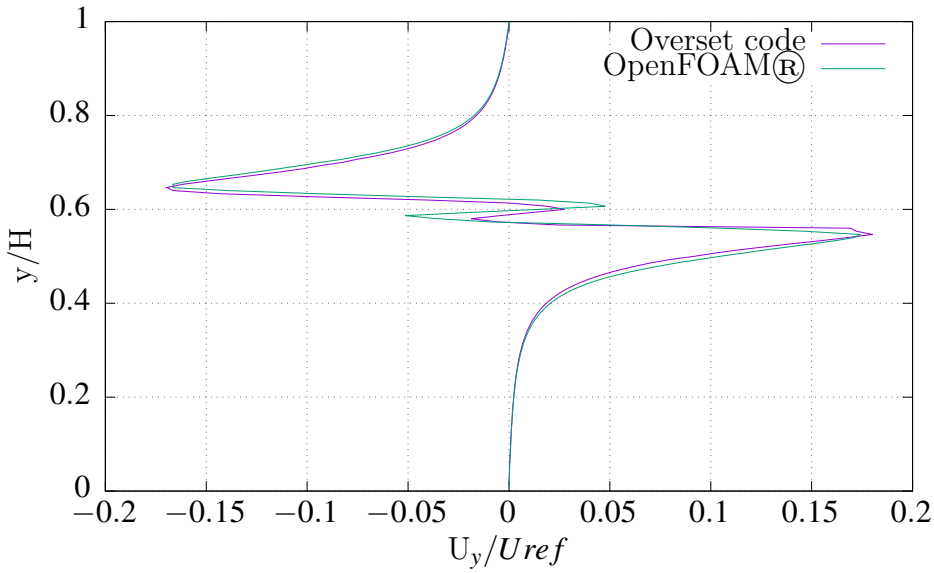

Figure 5.34: Vertical velocity profiles. $x = 16\ m$, $t = 9\ s$.

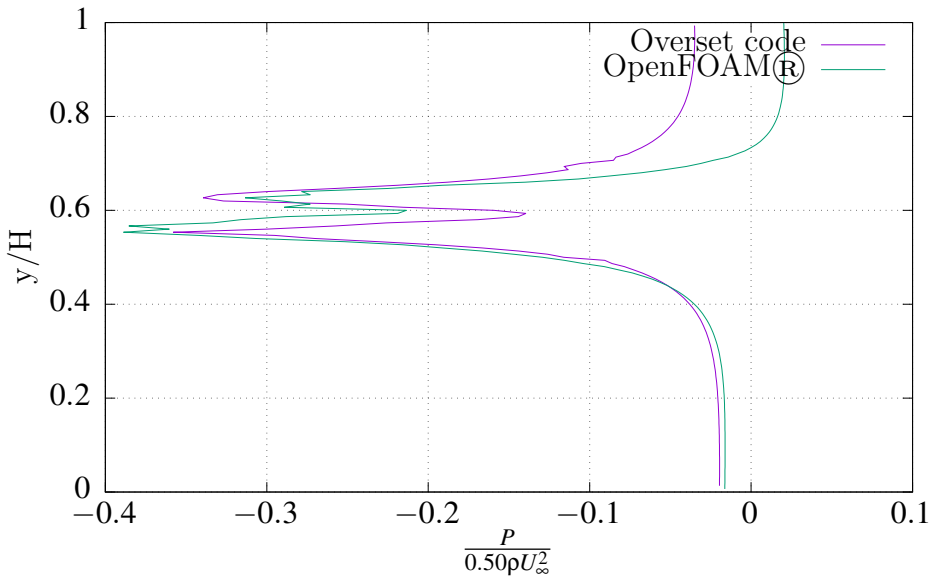

Figure 5.35: Pressure profiles. $x = 16\ m$, $t = 9\ s$.

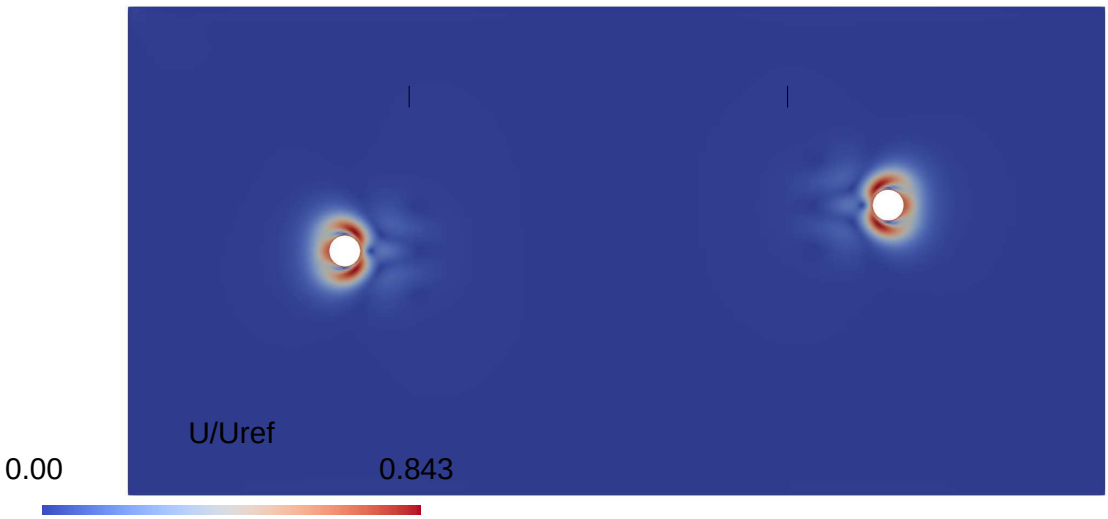

Figure 5.36: OpenFOAM® velocity magnitude field at $t = 15\ s$.

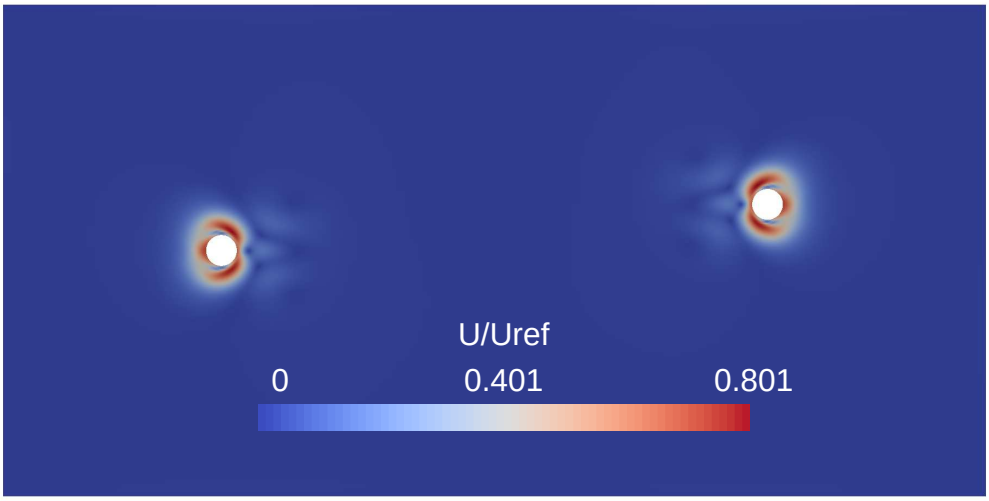

Figure 5.37: Overset code velocity magnitude field at $t = 15\ s$.

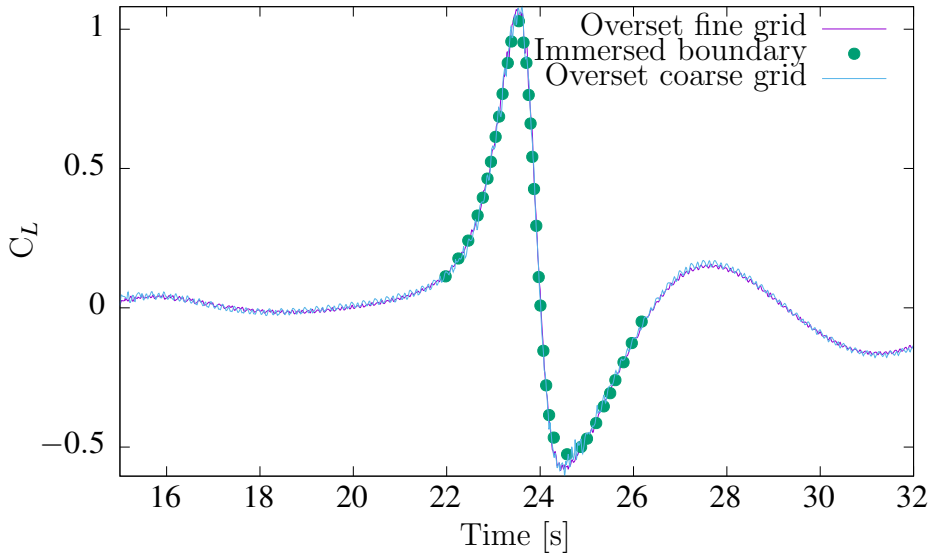

Figure 5.38: Lift coefficient of the upper cylinder.

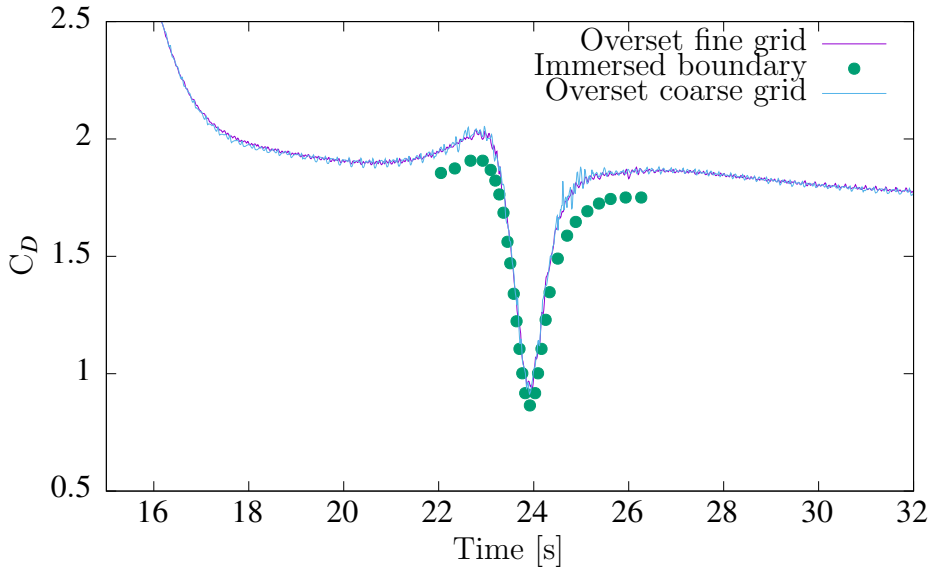

Figure 5.39: Drag coefficient of the upper cylinder.

5.4 General Planar Motion Plates

For this last exercise a more complex solid body motion scenario is proposed. Translational and rotational movement take place simultaneously for two rectangular plates enclosed in a square cavity filled with a fluid at rest. Figure 5.40 is an outline of the computational domain and grids, which comprises: a square cavity of 4 *m* per side discretized with a uniform background grid of 100×100 cells, while each rectangular plate has a width of 0.50 *m* and a height of 5 *cm*. The left hand side bar has its centre of mass located 40 *cm* off the origin in the negative horizontal direction, with the right hand side plate located 40 *cm* to the right of the origin. Each rectangular plate was surrounded by four block structured grids in the overset code as shown in Fig. 5.41 totalling 5614 cells. It is important to mention that the structured nature of the overset code solver requires any given grid to be either a single hexahedron, or a set of hexahedra coupled together to discretize complex geometries. For this particular case the grid around the plate could have been a single C-shaped hexahedron sacrificing grid quality around the corners, or as it was done, a set of hexahedra to preserve grid quality around the sharp corners. As it can be seen in Fig. 5.41, the mesh is not uniform and has some degree of refinement around the four corners. On the other hand, the unstructured solver nature of OpenFOAM® does allow to construct the grid around the plate using a single block whilst preserving good-cell quality, therefore the OpenFOAM® mesh for each plate is a single uniform grid as shown in Fig. 5.42 consisting of 5540 cells. It becomes clear from inspecting both previous figures that the body fitted grids for each code are not exactly the same, especially around the corner and plate height areas where the overset code exhibits more resolution, and along the width of the plates, where the OpenFOAM® grid is finer. It is obvious that these differences may have an impact on the the flow field computation in those areas, however, as will be demonstrated in the results section, they do not warrant adapting the OpenFOAM® mesh since both solutions compare satisfactorily.

Initially there is a 90 ° offset in the relative orientation of the plates, and each plate is subject to general planar motion. The compound motion for the left and right hand side

plates is given respectively by

$$Leftplate = \begin{cases} y = 0.20\sin(2\pi t)\ [m], \\ \omega_z = -2\ \frac{rad}{s}, \end{cases}$$

$$Rightplate = \begin{cases} y = -0.20\sin(2\pi t)\ [m], \\ \omega_z = 2\ \frac{rad}{s} \end{cases}$$

comprising oscillatory linear translation in the vertical direction, and rotation about the centre of mass of each body with constant angular velocity in opposite directions. The Reynolds number based on the maximum linear velocity at the edges of the plates exclusively due to rotation ($U_\infty = \omega r$), and the width of the plates is $Re = 400$. The initial velocity of the centre of mass of each plate is computed based on the oscillatory linear motion kinematics. Once the plates are set in motion they are allowed to complete 2.5 cycles of linear motion in 2.5 s.

Both codes employ the Backward time discretization scheme with a $\Delta t = 5 \times 10^{-04}\ s$ which ensures the Courant numbers are always less than 1.

Due to the compound motion, the initial positioning of the bodies, and the range of the body fitted grids, there are instants when both rectangular grids intersect (see Figures 5.43, 5.44). Although not to the same extent as in the previous case, there were times when the complex overlapping structures allowed the hole cutter algorithm to activate cells which were unused in either the previous or two previous time steps. As also noted in previous cases, the modified interpolation implementation always succeeded in providing the required values for the current time computation in those cells. Static Dirichlet boundary conditions for the momentum equations were established for the cavity walls, while the corresponding time-dependent values for each cell belonging to the rigid boundaries of the plates were calculated according to rigid-body kinematics (note that each wall cell has a different velocity due to combined translation and rotation).

The evolution of the vorticity field from $t = 0.1\ s$ to $t = 0.60\ s$ is depicted in a sequence of images encapsulated in Figures 5.45, 5.46, and it serves to demonstrate the complex flow patterns emerging as a result of the general planar motion of the plates. Initially

vortices generated close to the outer extremes of both plates are advected towards the cavity walls, while vortices formed near the inner extremes of the plates interact with each other and their trajectories depend heavily on the motion of the plates. This situation is not as clear as time progresses and the vortical interactions become more complex.

The first comparison with the OpenFOAM® solution is the velocity magnitude profile along a vertical line covering the entire length of the cavity located at $x = -0.750\ m$, its location and grids at a simulation time of $0.8\ s$ are shown in Fig. 5.47. This location was selected because at a height of about $2/3$ of the cavity length, the first vortex being shed off the bottom left surface of the left hand side plate can be found (refer to Fig. 5.50). As shown in Figure 5.48 both the peak velocity magnitude value and its location are in good agreement for both solutions.

Generally speaking, the flow induced by the movement of both plates is complex, and it is more appropriate to compare the flow field throughout the key regions in the computational domain to assess the quality of both solutions. Figures 5.49 through 5.54 do precisely this via the velocity magnitude field for both codes. In terms of maximum values the worst case scenario is for the first set of images (5.49, 5.50) with the difference being $\approx 3.70\%$ and the overset code always predicting higher velocities. By inspecting the location of these peaks it can be realised that they are mainly located around the sharp corners of the plates where the overset code has more grid resolution. Notably both solutions also agree in terms of the location of the main flow features, especially in the regions near both plates. The most notable differences are observed in the regions far from the plates where the background grid is actually exactly the same. This is not intuitive and it might be related to the differences in the inter-grid interpolation algorithms.

Considering alternative techniques to solve this problem, the grid-deformation approach can be ruled out quickly due to the presence of more than one solid body along with the complex rigid-body motion of the plates. Its use would severely distort the grid which would most likely result in solver divergence. If an alternative boundary-conforming grid methodology needs to be used, then the only reasonable choice would be to regenerate the near-body mesh each time step. Apart from the computational overhead inherent to this procedure, interpolation of the flow field variables would still need to take

place from the old grid to the new mesh in order to solve the time-dependent governing equations. The use of this approach is difficult to justify because it is precisely the interpolation procedure one would ideally want to erradicate from the solution when compared to an overset-grid methodology.

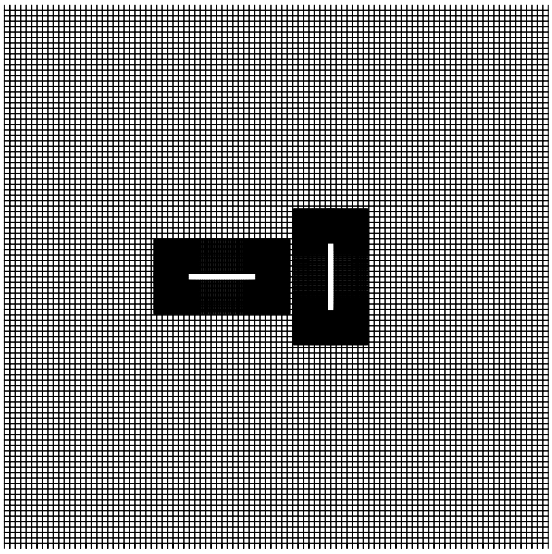

Figure 5.40: Initial configuration and grids for the mixer problem.

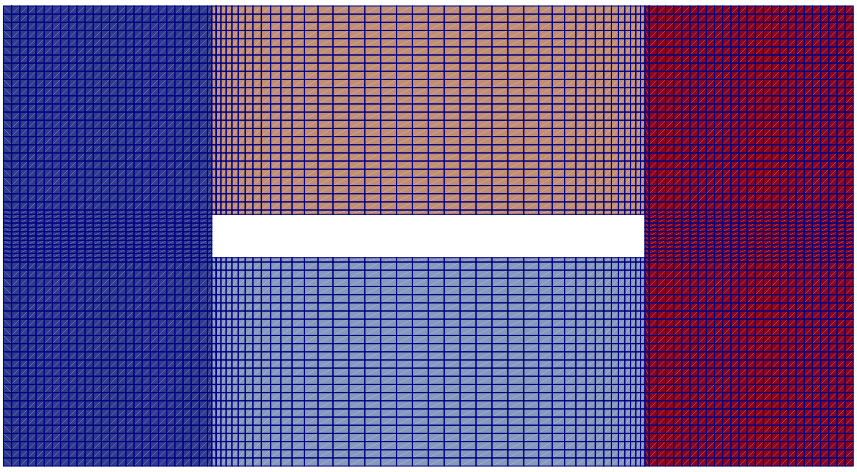

Figure 5.41: Grids employed to discretize the flow around a rectangular plate for the Overset code.

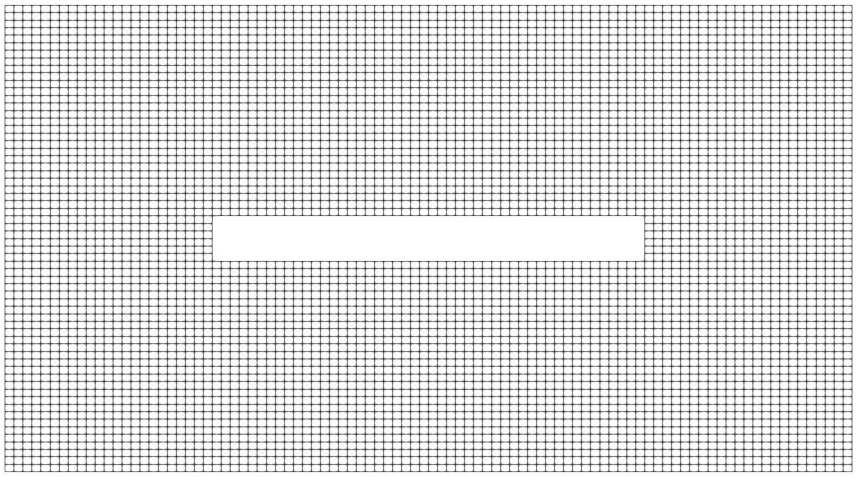

Figure 5.42: Grid employed to discretize the flow around a rectangular plate in OpenFOAM® .

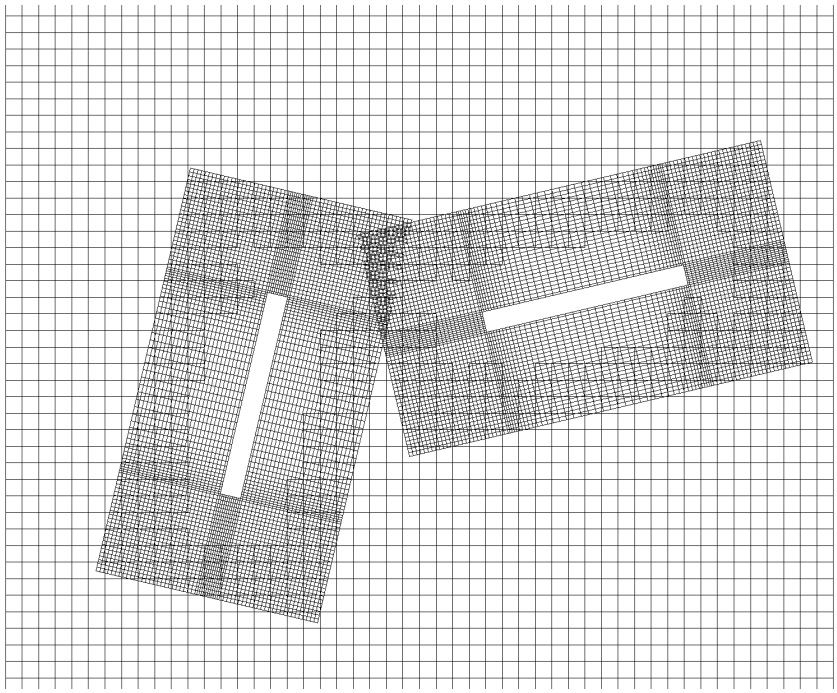

Figure 5.43: Overlapping zones at $t = 0.8\ s$.

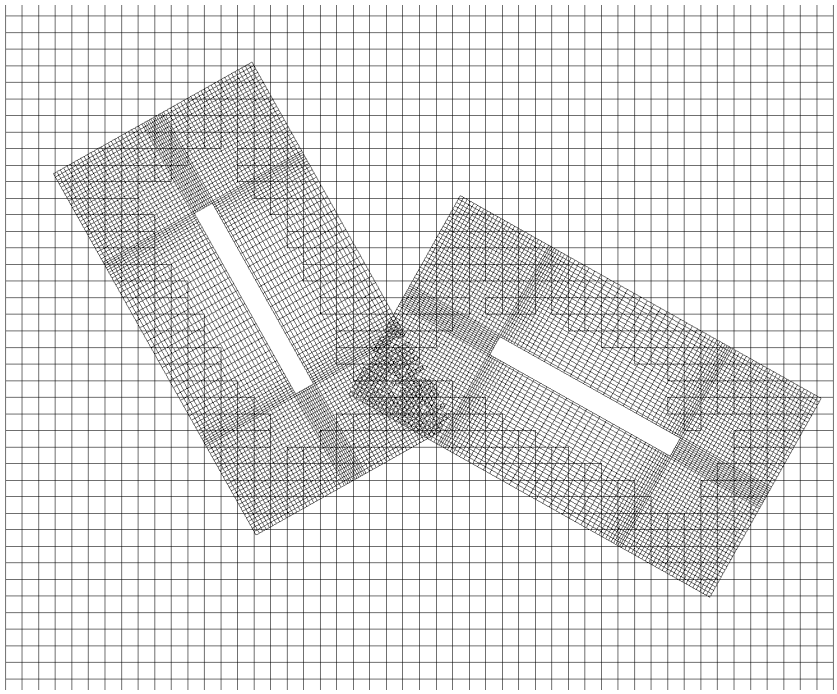

Figure 5.44: Overlapping zones at $t = 2.1\ s$.

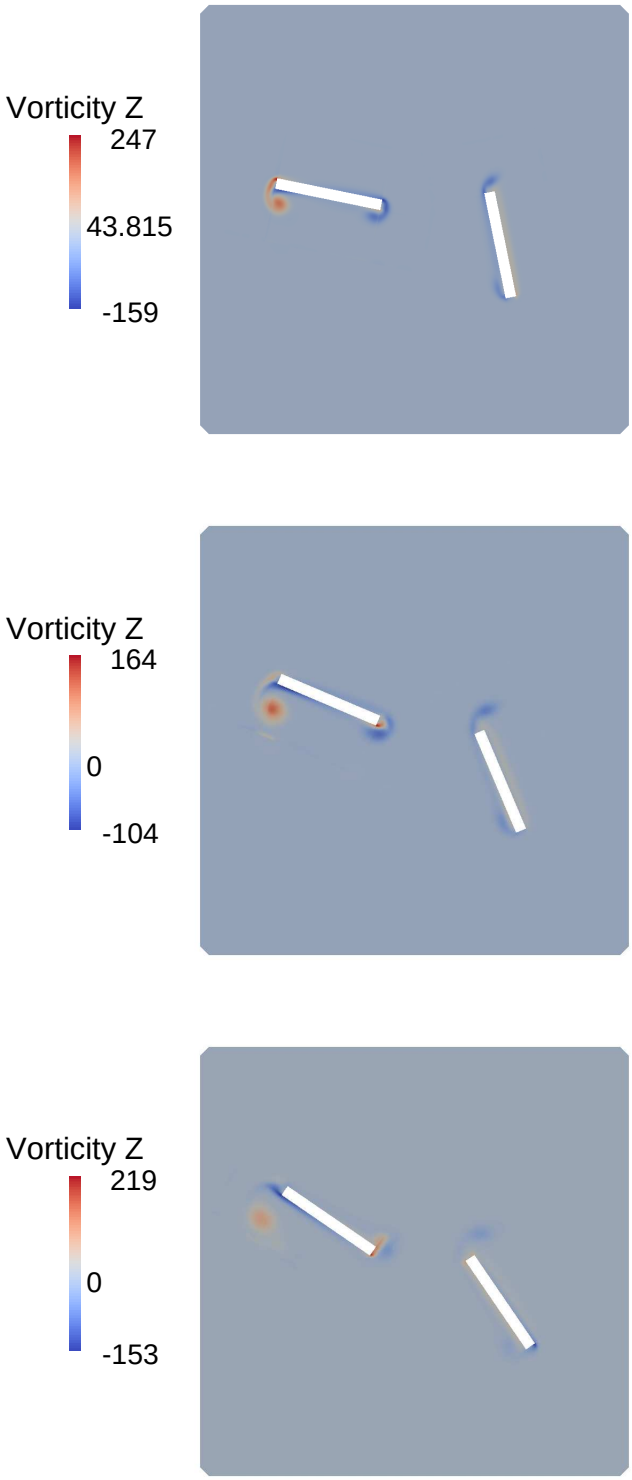

Figure 5.45: Evolution of the vorticity field 0.1 s (top), 0.2 s (middle), 0.3 s (bottom).

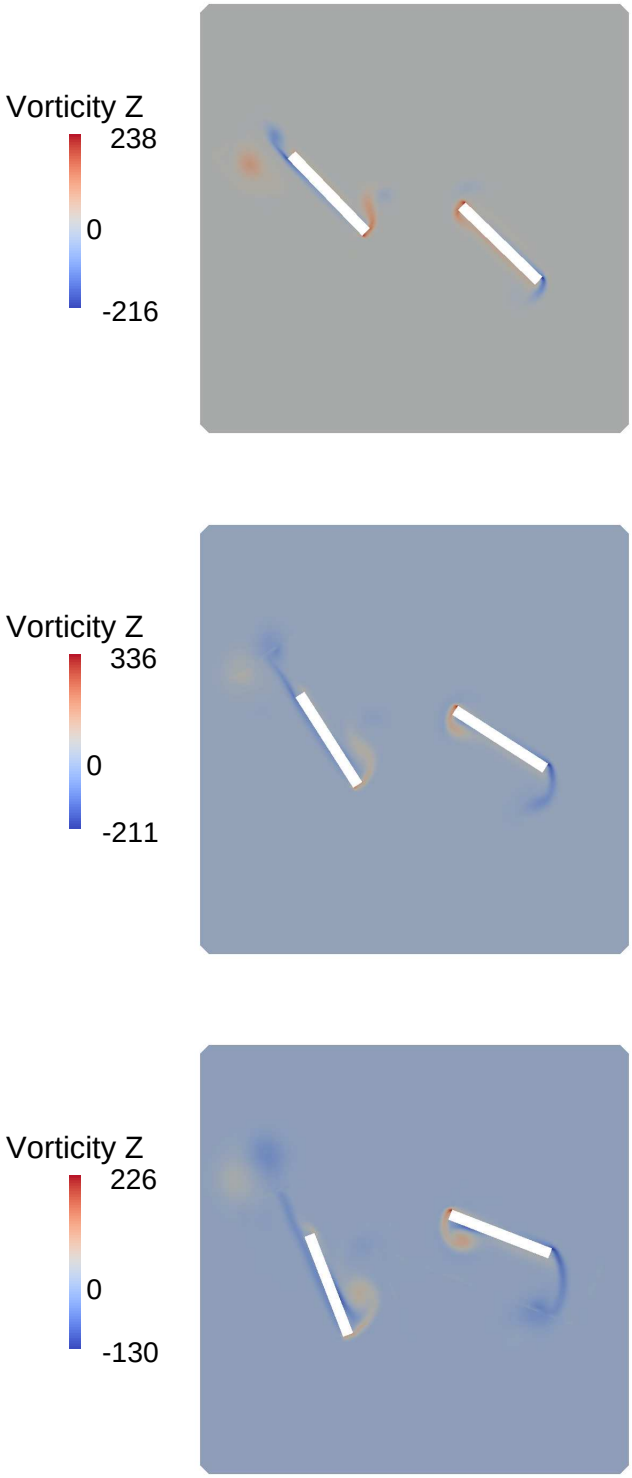

Figure 5.46: Evolution of the vorticity field continuation 0.4 s (top), 0.5 s (middle), 0.6 s (bottom).

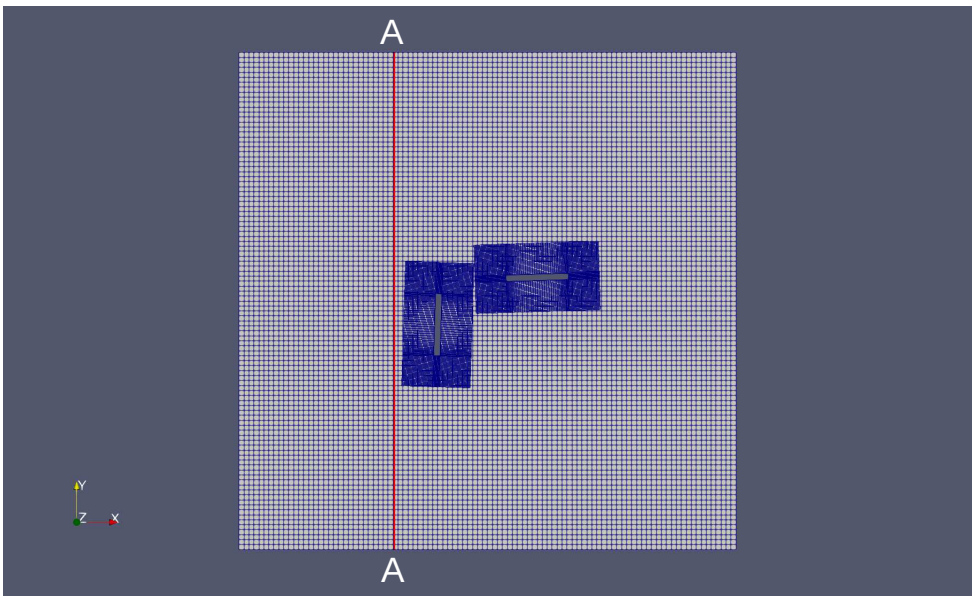

Figure 5.47: Grids and selected profile location for comparison with OpenFOAM® at $t = 0.8\ s$, $x = -.750\ m$

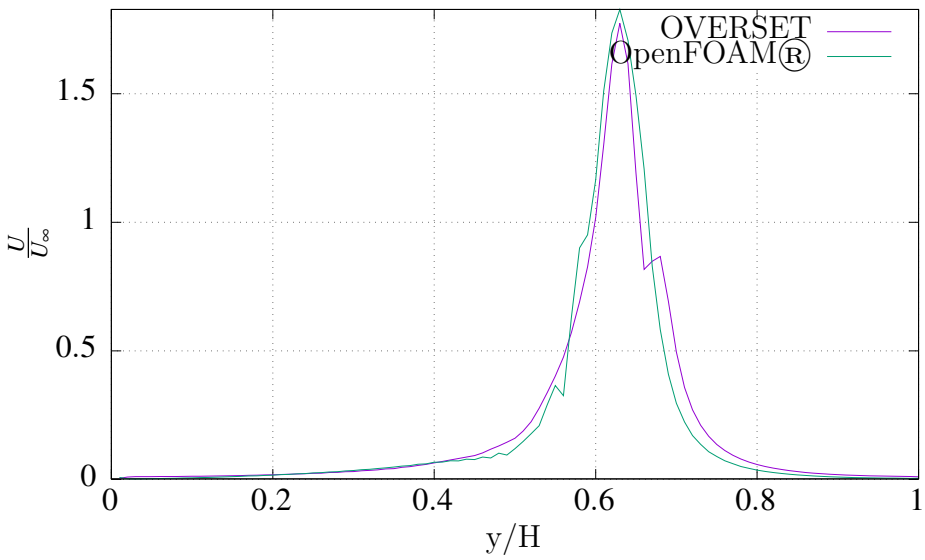

Figure 5.48: Velocity magnitude comparison at $t = 0.8\ s$, $x = -.750\ m$

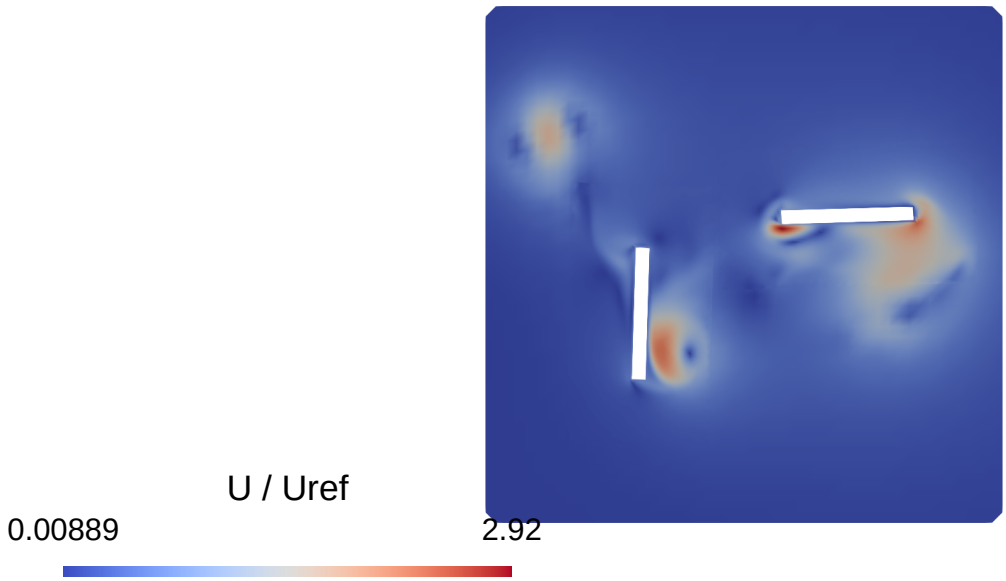

Figure 5.49: OpenFOAM® velocity magnitude field at $t = 0.8\ s$.

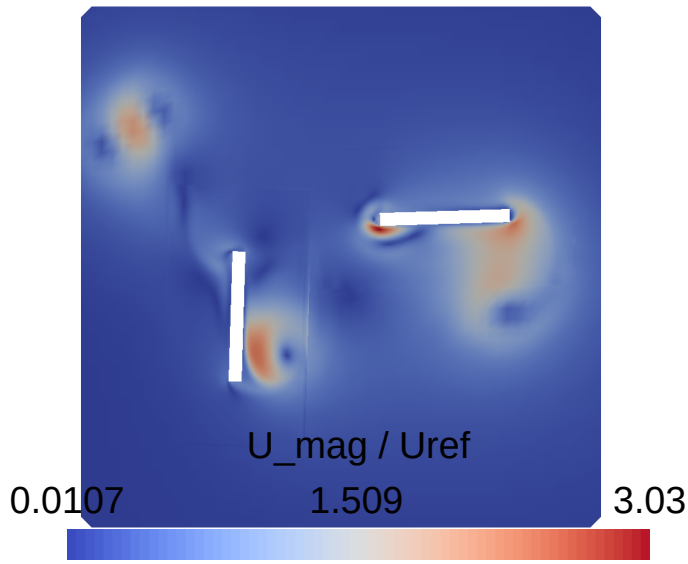

Figure 5.50: Overset code velocity magnitude field at $t = 0.8\ s$.

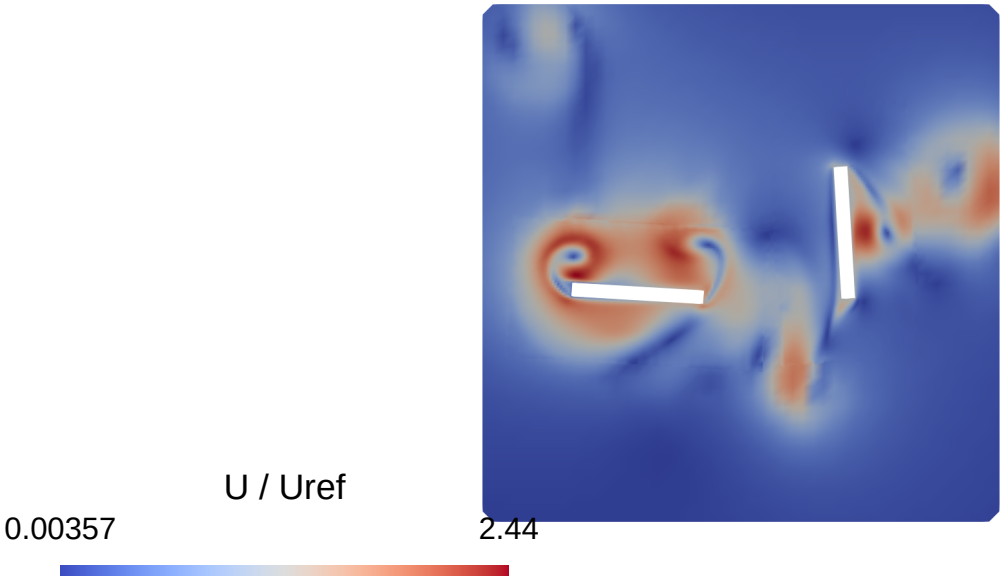

Figure 5.51: OpenFOAM® velocity magnitude field at $t = 1.6\ s$.

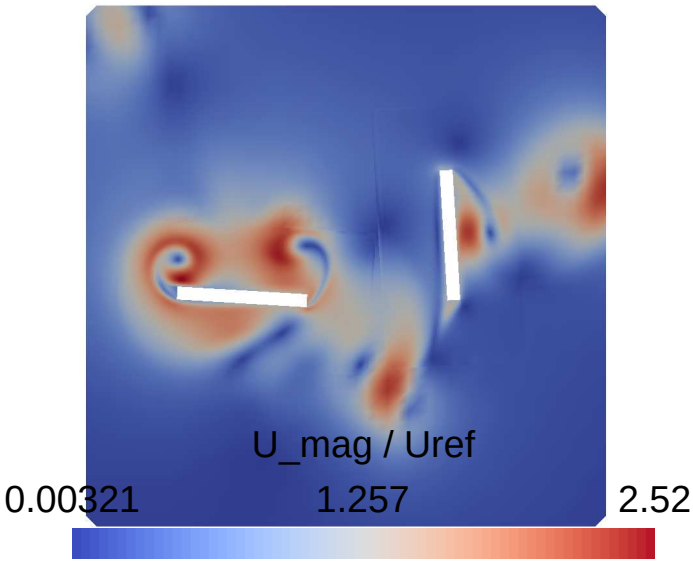

Figure 5.52: Overset code velocity magnitude field at $t = 1.6\ s$.

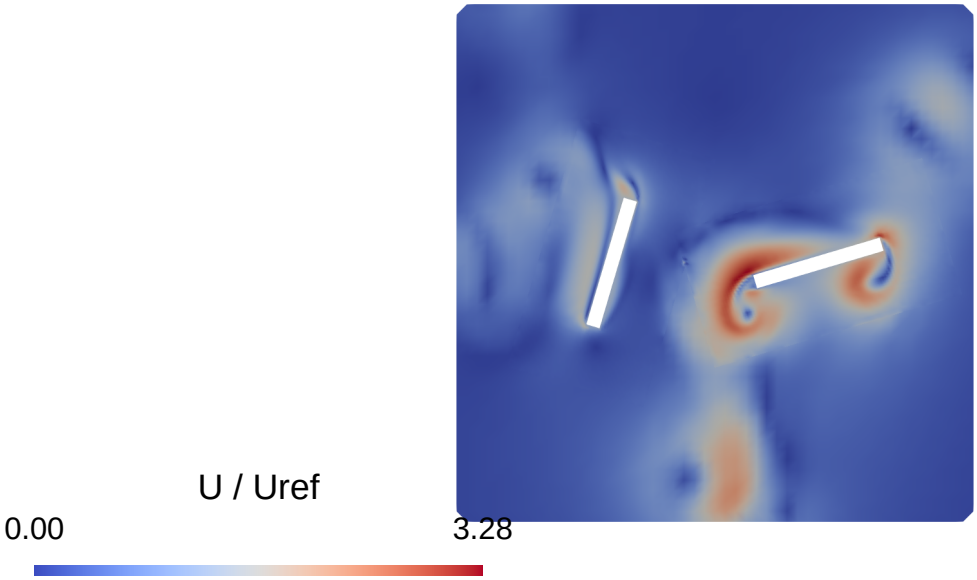

Figure 5.53: OpenFOAM® velocity magnitude field at $t = 2.5\ s$.

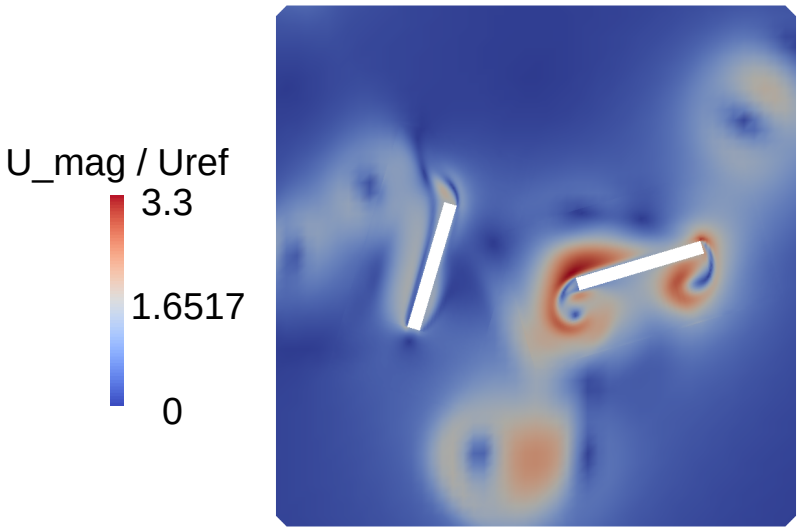

Figure 5.54: Overset code velocity magnitude field at $t = 2.5\ s$.

5.5 Closure

The comparison between the newly released overset functionality of the OpenFOAM® solver, and this in-house implementation of the overset-grid method revealed great agreement for the set of problems with varying complexity presented in this chapter. Results showed the potential intrinsic differences between the hole cutting algorithms and inter-grid communication procedures did not have a significant impact on the qualitative features of the flow field, and considering the exact same initial grids were not used in both codes, the degree of quantitative consistency in terms of the extrema of the flow field is somewhat surprising. Implicitly this comparison served to validate the one-way FSI coupling algorithm implemented in the overset code, and verified the rigid-body solver under a scenario that was not presented in the previous chapter: general planar motion.

Chapter 6

Two-Way FSI Simulations

6.1 Introduction

The main objective of this Chapter is to validate the overset code for two-way rigid-body FSI simulations. This is achieved via two problems for which the motion of the solid is not prescribed, and so there exists a a reciprocal dependency amongst the kinematic properties of both media.

In the first case, the solution of a free falling cylinder under the gravitational field is compared with another numerical approach, and traverse vortex induced vibrations (VIV) of a circular cylinder exposed to a laminar stream are explored afterwards. Although not as popular in the literature as its turbulent flow regime counterpart, there are plenty of numerical studies that have tackled this problem. More importantly, experimental data in the form of oscillation amplitudes and frequencies for a range of laminar Reynolds numbers are available and will serve to benchmark the current overset code implementation.

Lastly, a case study regarding the fluttering motion of rectangular plates under the influence of gravity is proposed and explored. Throughout all cases some important numerical aspects inherent to the partitioned coupling algorithm will be discussed, such as relaxation and the degree of coupling required to obtain satisfactory solutions.

6.2 Falling Circular Cylinder

The first two-way FSI case entails the free fall of a cylinder subject to gravitational forces and immersed in an initially static fluid. This problem was solved numerically in [89] using a vorticity based particle solver. Fluid viscous and pressure forces are solely responsible to counter the gravitational field. The initial phase is marked by a sharp linear increase in the cylinder's downward momentum due to the gravitatinal force. This scenario is short lived as fluid resistance also builds up swiftly (being a function of velocity), and eventually it is responsible for limiting the object's speed to a terminal value. It is at about this time that periodic vortex shedding develops, and as a result oscillating fluid forces cause the cylinder's velocity to exhibit oscillating behaviour as well.

Since the dimensions of the enclosing cavity were not reported in the source, a square cavity of 10.4 *cm* per side as shown in Fig.6.1 served as the background grid for this simulation. Additionally, an intermediate background mesh has been employed as a transition computational domain between the coarse cavity grid, and the fine body fitted mesh around the circular cylinder. The need to include this intermediate grid arises due to a significant difference in the dimensions of the cavity and the moving body. Otherwise an excessively fine mesh would have to be generated for the background grid to give sufficient overlap with the body fitted mesh. Or alternatively, the body fitted grid would have to be rather coarse in order to have similar sizing to the coarse background mesh in the overlapping region.

Thus, the objective of the intermediate grid is to serve as an interface between regions of low and high grid resolutions. Furthermore, the intermediate grid can be optimised in terms of size by covering solely the areas where the solid body is expected to be located throughout the length of the simulation. For static and one-way FSI cases this is, or can be, known a priori. However, for a general planar motion two-way FSI problem the range of motion of the body is not known beforehand, and optimising the intermediate grid can be difficult. A simple yet powerful solution is to have the intermediate grid follow closely the solid body by setting its forcing term to mimic the one belonging to the actual solid body. This can be done in several ways; for this problem the previous time forcing term of the solid body is employed to move the intermediate grid in the current time step. The

intermediate grid's centroid is positioned slightly above the actual solid body. Fig. 6.2 illustrates the initial grid configuration *before* the hole cutting algorithm sets the active and inactive regions. Notice how part of the intermediate grid lies outside of the cavity domain. This is not an issue for the hole cutting algorithm as long as there is enough overlap between both cavity and cylinder solid walls. Figure 6.3 shows a zoom of the region of interest once the hole cutting algorithm has set grid priorities. The number of cells for the background, intermediate, and body fitted grids are: 14400, 9230, and 5445 respectively.

The cylinder has a diameter of 1.6 *mm* and its centroid is initially positioned at the origin of the coordinate system. The solid to fluid density ratio is 2, and the Reynolds number based on the Galilei number is $R_e = \sqrt{\left(\frac{gD^3}{\nu^2}\right)} = 200$. By employing a time step of 1×10^{-04} s both advection and grid motion Courant numbers are always kept under 1. Dirichlet boundary conditions are set for the momentum equations on the walls: the values being zero for the static cavity walls and the instantaneous centre of mass velocity for the cylinder surface (the cylinder motion is restricted to the vertical direction as per the original article). The partitioned coupling algorithm performs one predictor step and between one and three correctors depending on the FSI coupling residual, which in this case monitors the diameter-normalised centre of mass location for two consecutive correctors as $FSI_{residual} = |Y_{current} - Y_{previous}|$. For this particular problem the solution was obtained without FSI relaxation.

During the very first instants of the falling motion the cylinder attains its maximum descending acceleration, since the opposing fluid forces are at a minimum as seen in Fig. 6.5. As the cylinder further advances downward, two stretching vortices are formed behind its trailing surface. It is worth noting that just before these vortices detach there is an additional increase in the opposing fluid force, which is responsible for an additional reduction of the cylinder terminal velocity. The onset of periodic vortex shedding is depicted in Fig. 6.4 through the velocity magnitude field. This process is delayed until at about 0.40 s, and can be confirmed by the lift coefficient oscillations illustrated in Fig. 6.5.

The comparison of both numerical solutions is plotted in Figs. 6.6, and 6.7. There is no visible difference for the centre of mass position and the maximum difference of

the centre of mass velocity is around 3%. The original source only reports results up to about 0.30 s, just before the onset of periodic vortex shedding. This might be indicative of limitations in the original methodology to resolve the fluid-solid interaction properly when the coupling becomes more complex. Even though both (this and the original research article) FSI coupling strategies are similar (partitioned predictor-corrector FSI approaches), vorticity-based fluid flow solvers are known to require special procedures to enforce the no-slip kinematic constraint correctly on solid boundaries [90]. This is also reflected in [89], and the described FSI-coupling requires an iterative algorithm to remove an emerging non-zero spurious slip velocity at the solid-fluid interface. It is expected for this additional difficulty to further contribute to the complications encounterd by partitioned FSI algorithms to properly resolve strongly-coupled interactions due to added-mass effects.

This test case served to validate the current implementation for two-way FSI problems with relatively low solid-fluid mass ratios, which are known to cause solver instabilities. The use of a 'motion-tracking' intermediate grid for two-way overset FSI cases allows to effectively direct mesh refining efforts to critical regions.

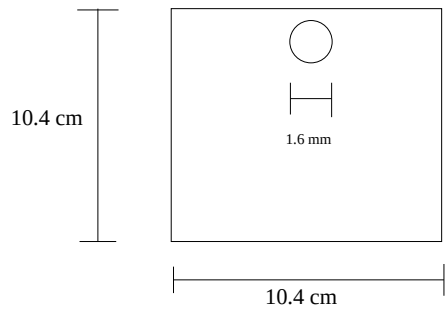

Figure 6.1: Outline of the falling cylinder fluid domain.

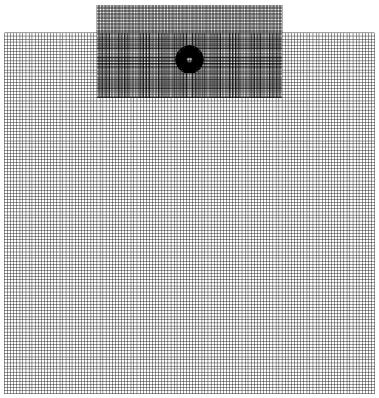

Figure 6.2: Computational domain for the free falling cylinder.

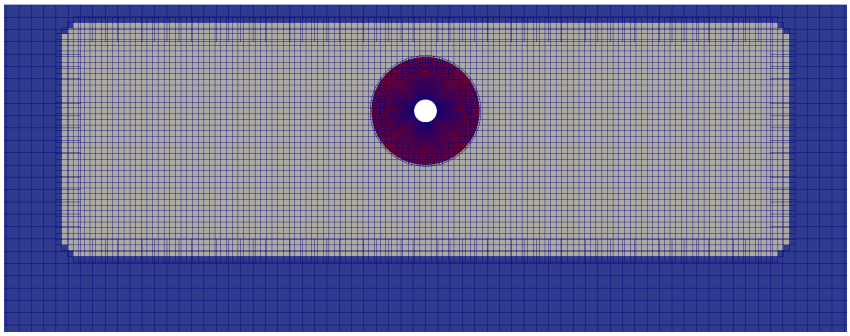

Figure 6.3: Zoom of the computational domain once holes have been cut.

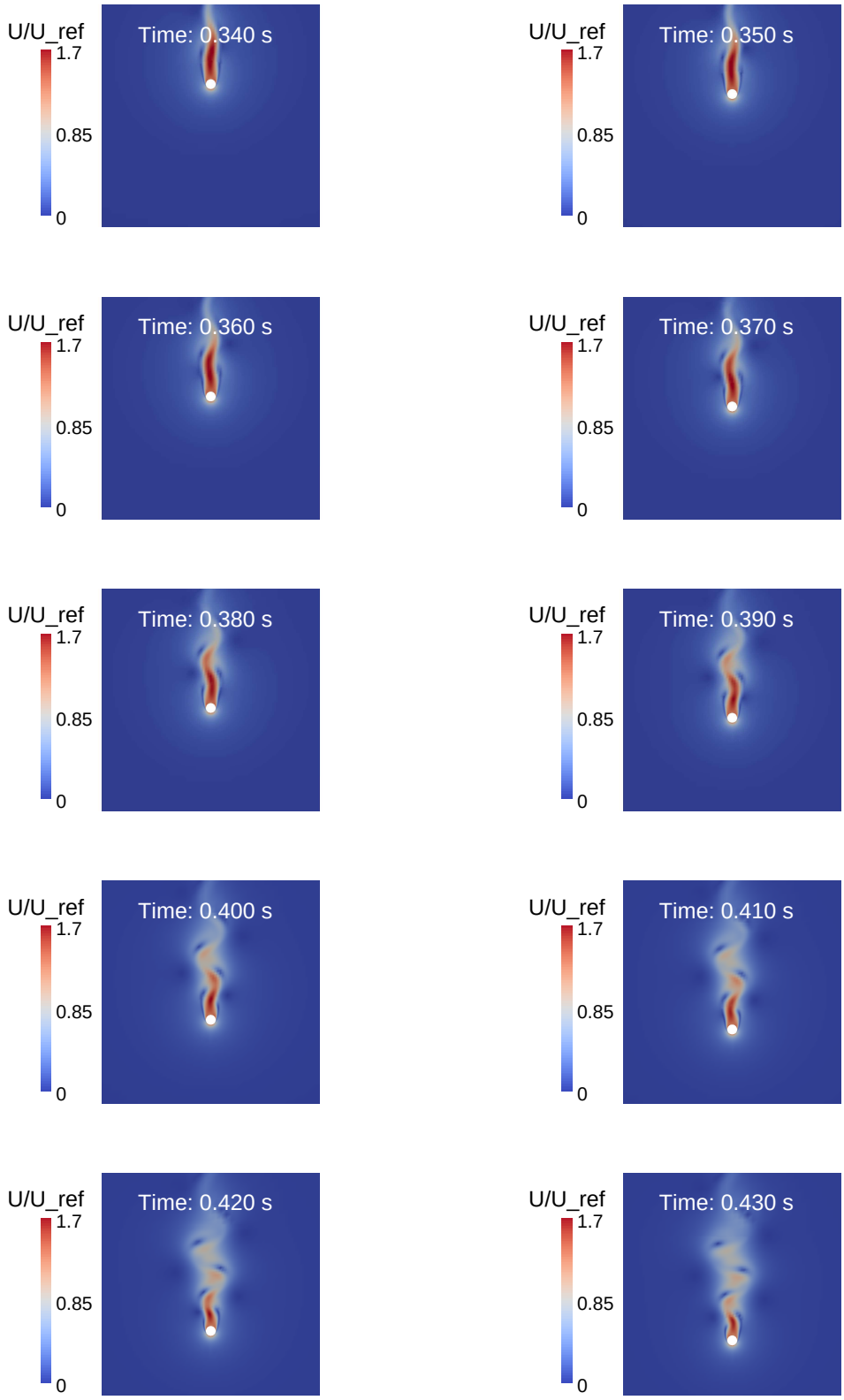

Figure 6.4: Evolution of the falling cylinder velocity magnitude field. Left to right, top to bottom. From $t = 0.34\ s$ to $t = 0.43\ s$.

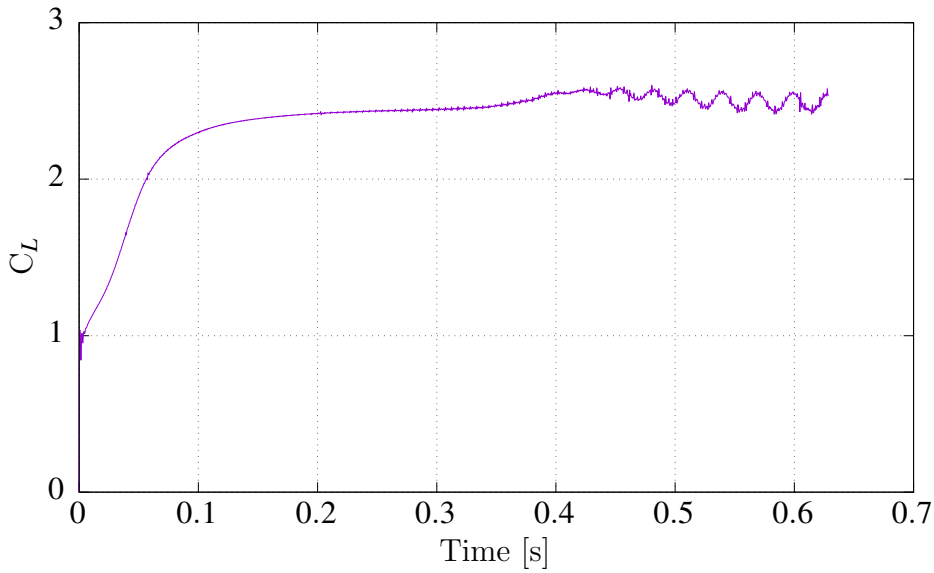

Figure 6.5: Evolution of the lift coefficient.

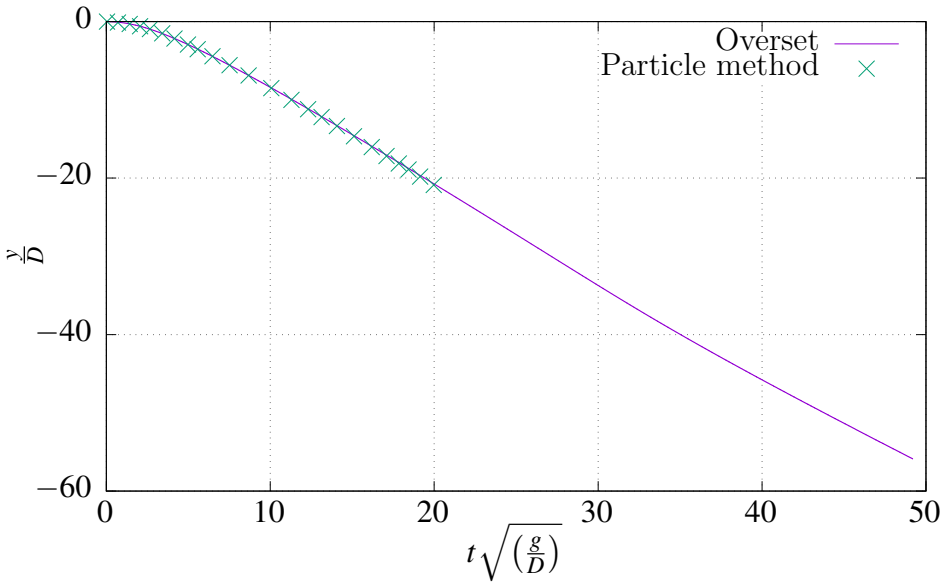

Figure 6.6: Cylinder vertical position transition.

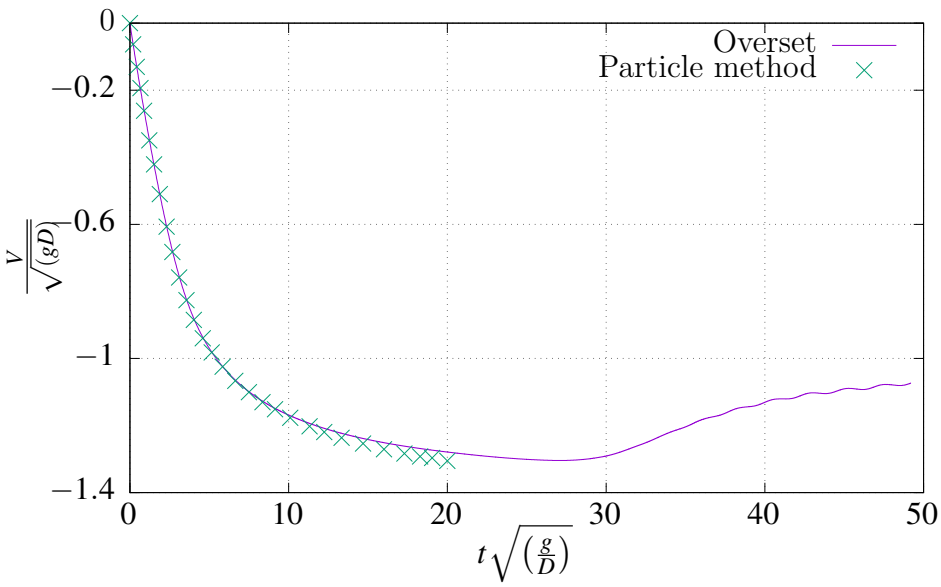

Figure 6.7: Cylinder vertical velocity transition.

6.3 VIV of a Circular Cylinder

This section describes the modelling of a fluid-mechanical system comprising a mass-spring-damper structure with the objective of capturing critical flow induced vibrations. An elastically mounted cylinder (refer to Fig. 6.8) paired with a viscous damper is exposed to a free stream entering through the left side, whose velocity varies according to the Reynolds number based on the cylinder diameter in the range $94 \leq R_e \leq 118$ (totalling 7 different cases). The mechanical properties of the linear spring and the viscous damper are $k = 5.79\ [N/m]$, and $c = 0.325\ [\frac{g}{s}]$. Both mechanical elements allow the cylinder to oscillate exclusively in the vertical direction. The mass of the cylinder is $m = 2.979\ g$ and the corresponding solid to fluid mass ratio is 149.10.

The subject of fluid-induced vibrations has been long studied due to the potentially catastrophic consequences it can have for humans, or engineering equipment. Particularly, VIV of a circular cylinder have been extensively explored for high Reynolds numbers as summarised in [91] . This reference is an excellent publication revealing many of the influencing factors such as the existence of a critical mass, and the resulting vortical structures under free or forced cylinder loading conditions. It also presents a broad literature review on the subject, classifying numerical and experimental work according to the range of Reynolds numbers. In such classification the only laminar experimental work is the one by Anagnostopoulos et al. [92], which ultimately provides the data to validate the current two-way FSI overset implementation. Shortly thereafter several numerical studies emerged to gauge different methodologies (Immersed Boundary, Finite Element) devised for FSI predictions with the available experimental data. One of those studies is the one by Samaniego et al. [93] in which the authors employed the Finite Element method to capture the 'lock-in' resonant behaviour experienced in the experiment. The geometrical parameters and physical properties used in this work are those reported in [93].

Dirichlet boundary conditions for the momentum equations apply for the left hand side flow entrance surface in Fig. 6.8, and for the cylinder surface. For both top and bottom surfaces a zero gradient in the normal direction, and zero normal velocity are imposed to represent slip walls. The flow exits the computational domain through the rightmost

surface, for which zero gradients in the normal direction are set for the momentum equations. The exit is placed ≈ 18 diameter lengths behind the cylinder and so the outflow boundary assumption is suitable.

The rectangular channel was discretized using 5056 and 20224 cells for the coarse and fine grids respectively. A single C-shaped structured block was wrapped around the cylinder surface and discretized with 660 and 3159 cells for the coarse and fine grid configurations respectively. Figure 6.9 shows the coarse grid configuration, whilst Figure 6.10 is a zoom of the respective overset cylinder mesh. The backward time discretization scheme was used for all simulations, using time steps that keep the advection Courant number close to 1 for the maximum amplitudes experienced ($\Delta t = 5 \times 10^{-03}$ s coarse grid, $\Delta t = 1 \times 10^{-03}$ s fine grid).

The effect of grid resolution on the resulting oscillating amplitude for the case $R_e = 102$ can be seen in Figure 6.11. This particular Reynolds number was selected for mesh comparison because it is the onset of a series of critical R_e numbers for which the cylinder experiences the most extreme vibrations. This is clearly the case for the fine grid, as a steady increase in the maximum amplitude can be appreciated after each vibration cycle. Although the coarse grid results eventually (slightly visible only after 60 s, not shown) display signs of this behaviour, the maximum amplitude in each oscillating cycle appears to be constant during the initial phase.

A series of images portraying the evolution of vortex shedding via velocity magnitude contours is shown in Fig. 6.12. However, for this kind of problems it is more instructive to analyse the phenomenon by means of the cylinder response in terms of vibration amplitude range. The occurrence of 'lock-in' is clearly captured for all cases in the range from $R_e = 94$ to $R_e = 118$ as shown through Figs. 6.13-6.19.

In this set of images two sub-figures per case reveal the initial vibration modes (left), and the complete cylinder response (right) throughout the time interval. These figures, along with plots for the ratio of vortex-shedding frequency to natural frequency (Fig. 6.20), and cylinder maximum amplitude (Fig. 6.21) as functions of R_e serve to explain the different stages of the 'lock-in' phenomenon. Starting at $R_e = 94$ the frequency ratio is

around 0.83, with the cylinder convincingly experiencing low amplitude steady oscillating behaviour; notice however that the initial vibrating mode is preserved throughout the duration of the cylinder motion (Fig. 6.13).

Increasing the inlet flow velocity up to $R_e = 98$ results in a frequency ratio of around 0.85, under this condition the cylinder oscillating amplitude increases slowly as a function of time (see Fig. 6.14). Rather peculiar vibrational modes where the cylinder progressively increases its maximum amplitude emerge for the next two flow regimes $R_e = 100$, and $R_e = 102$ (Figs. 6.15, 6.16). Not only does the maximum vibration amplitude increase considerably (up to 40% of its diameter at $R_e = 102$), but these vibrational modes exhibit two stages: one in which the cylinder goes through individual oscillating cycles characterised by significant variations in amplitude, and a final mode in which the oscillation amplitude remains constant.

The above mentioned individual oscillation cycles disappear for the next two Reynolds number regimes: $R_e = 108$, and $R_e = 114$. An interesting feature is the fact that even though the frequency ratio remains remarkably close to 1, the maximum oscillating amplitude starts to decay back to small amplitudes as the Reynolds number increases up to $R_e = 118$. This is an interesting finding which is not intuitive.

Fig. 6.20 also serves to compare the current overset code results with experimental data, and the numerical study presented in [93] which uses the Finite Element Method. Both numerical solutions agree reasonably well in terms of frequency ratio during the initial flow regimes. The current overset code predicts frequency matching only slightly earlier ($\approx R_e = 100$) than the Finite Element solution ($R_e = 102$). However, experimental data exhibits frequency matching starting at $R_e = 108$. For this and the next regime $R_e = 114$ both experimental data and the overset code fully agree. The Finite Element solution overpredicts the frequency ratio at the latter regime. Both numerical studies overpredict the ratio at the last regime of $R_e = 118$.

Another benchmark for the current overset implementation can be found in Fig. 6.21, where data for the maximum amplitude reached as a function of the Reynolds number is presented. In this case both numerical solutions agree remarkably across the board, but there are considerable differences with the experimental data, especially in the higher

Reynolds number range. This is expected, as the vortex shedding frequency in both numerical solutions departs from the natural frequency of the cylinder during that phase. Qualitatively though both overset and Finite element results exhibit the same features as the experimental data: a slowly increasing maximum amplitude during the early stages, followed by an abrupt increase in the middle stage, ending with a decreasing amplitude trend.

The iterative nature of the partitioned coupling algorithm naturally calls for exploring the required degree of coupling to obtain accurate simulation results, as well as investigating the introduction of 'interface' under-relaxation or over-relaxation which might be necessary in some instances to aid numerical stability, or desired to accelerate coupling convergence.

This particular problem required no 'interface' relaxation, as the solution never exhibited signs of instability. However, in order to explore a possible convergence speed-up for the coupling algorithm, the dynamic relaxation factor (unbounded) introduced in Chapter 3 and its bounded (between 0.10-1) variant were tested. In order to assess the convergence of the predictor-corrector coupling algorithm the 'FSI residual' is defined as:

$$Fsi_{residual} = \left| \frac{Current_{state} - Previous_{state}}{Previous_{state}} \right| \tag{6.1}$$

where as seen in 3.3.5 the 'state' of a rigid body comprises information regarding its position, orientation, linear, and angular momentum at every instant in time. In the above equation the 'Current' and 'Previous' identifiers could refer to either the corrected and predicted values respectively, or two consecutive corrections as it would emerge in advanced iterations of the coupling algorithm. Thus, for this case the vertical displacement and momentum 'FSI residuals' are monitored as a function of the iterations of the coupling algorithm. The FSI coupling convergence criteria for this case was 1×10^{-04}. If the criteria was not met after 6 corrections the algorithm advances regardless of the residual.

Figures 6.22, and 6.23 present comparisons of the FSI residual when using the bounded variant of the relaxation factor with the unrelaxed solution for a flow regime of $R_e = 108$ for a solution time of 3.5 s. This flow regime was selected due to the swift increase in cylinder oscillation amplitude which could lead to instabilities. As both figures reveal,

the unrelaxed solution needed less FSI iterations than the bounded relaxation factor implementation. Figure 6.24 presents the corresponding comparison when the unbounded version of the relaxation algorithm was tested. In this case the unbounded version allowed over-relaxation which is why it required less iterations than its bounded counterpart, however, it still required more iterations than the unrelaxed solution.

In order to inspect the degree of coupling necessary in the FSI algorithm for this problem, two additional tests were set up: Weak coupling, which only consists of the explicit predictor stage, and a single predictor-correction iteration. Since the backward time discretization scheme was used, the weak coupling test employs a first order Euler explicit predictor for the very first time step, and then switches to a second order explicit predictor for the remainder of the simulation.

Figure 6.25 compares the single cycle algorithm with the strongly coupled version in terms of cylinder displacement for a flow regime of $R_e = 102$, and reveals that there is virtually no difference by using a single predictor-corrector iteration in the early stages of the simulation for the coupling algorithm. As the cylinder amplitude increases the coupling between both media becomes more elaborate and requires more iterations, this is reflected towards the later stages of the simulation where both the period and amplitude are visibly different.

By employing a very weak coupling algorithm (only the explicit predictor step) simulation times decrease significantly, but as seen in Figure 6.26 at the expense of accuracy. There is an immediate phase shift in the cylinder response, and an overprediction in terms of amplitude. This situation exacerbates for longer simulation times, as the weak nature of the coupling algorithm entails the use of a second order predictor compared to the third order achieved by the combined predictor-corrector strong coupling approach. Additionally, the physical fluid-solid coupling under higher cylinder oscillations becomes more complex and a simple predictor coupling approach will not suffice to transfer information back to the fluid.

Finally, the computed lift coefficient using both weak and strong coupling approaches is shown in Fig. 6.27. Although the difference is clear throughout the time range, it is more striking towards the end of the simulation, where an increase in the lift coefficient

amplitude for the weak approach is responsible for the large displacements experienced by the cylinder.

The vast majority of numerical solutions for VIV problems in the literature are computed with Immersed Boundary or Finite Element methods. This exercise has demonstrated the capability of the current overset implementation to capture the complex 'lock-in' FSI phenonemon through comparisons with experimental and numerical data, and thus it is a viable alternative to the dominating methodologies. This kind of resonant behaviour FSI problems is especially problematic for loosely coupled solutions, as seen through the coupling tests performed in this section, therefore explicit weak coupling is not recommended.

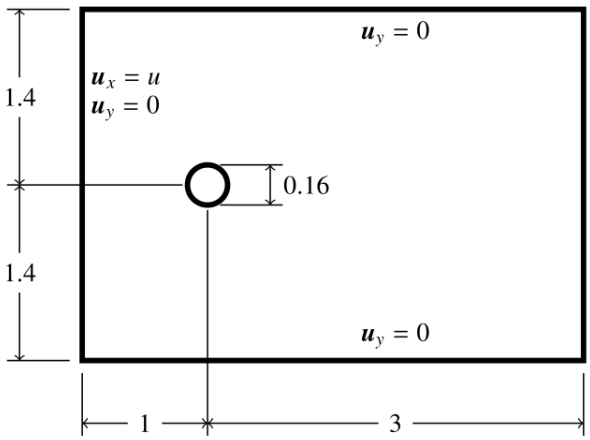

Figure 6.8: VIV Problem geometry, dimensions in [cm]. Image from[93]

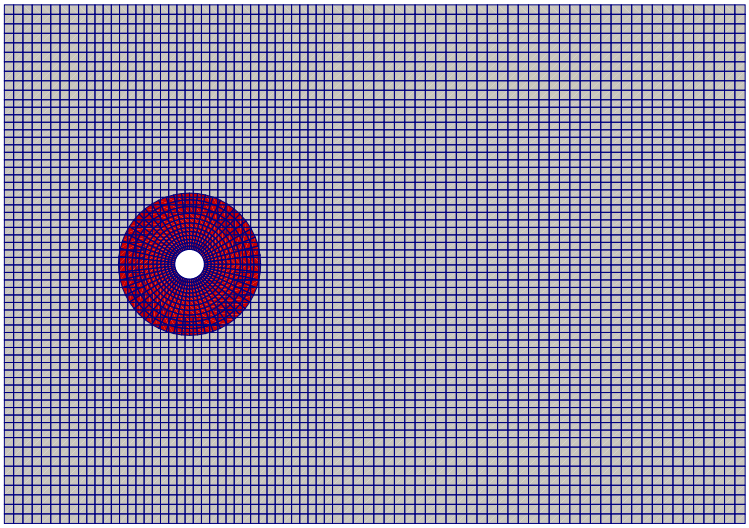

Figure 6.9: VIV Coarse mesh.

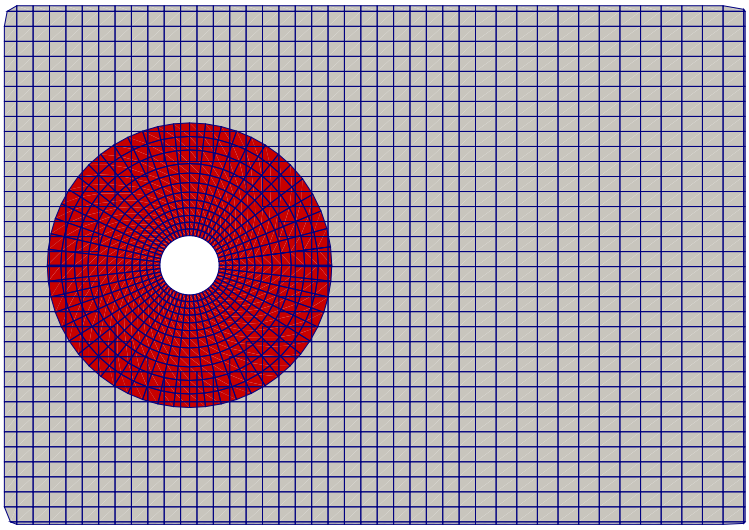

Figure 6.10: VIV Zoom of the overlapping grid section.

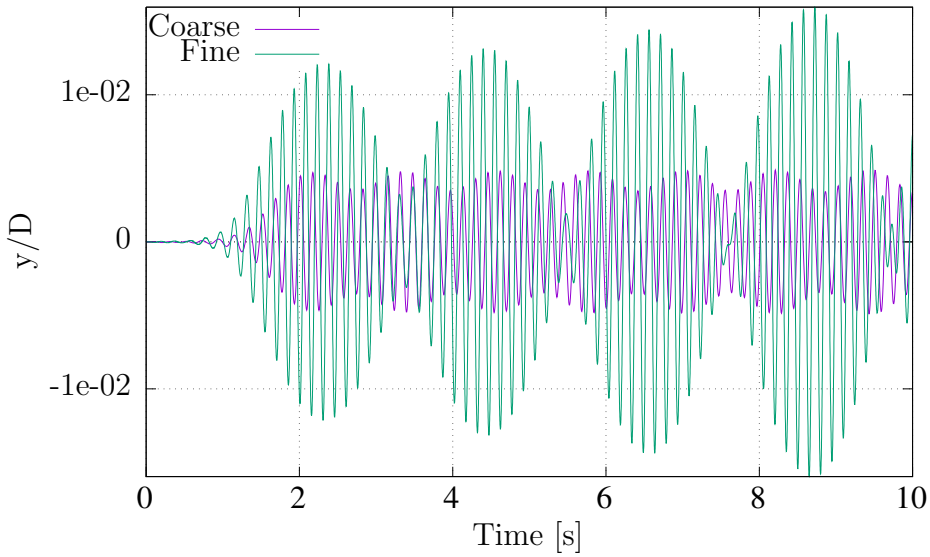

Figure 6.11: Oscillating amplitudes for the two grid resolutions at $R_e = 102$.

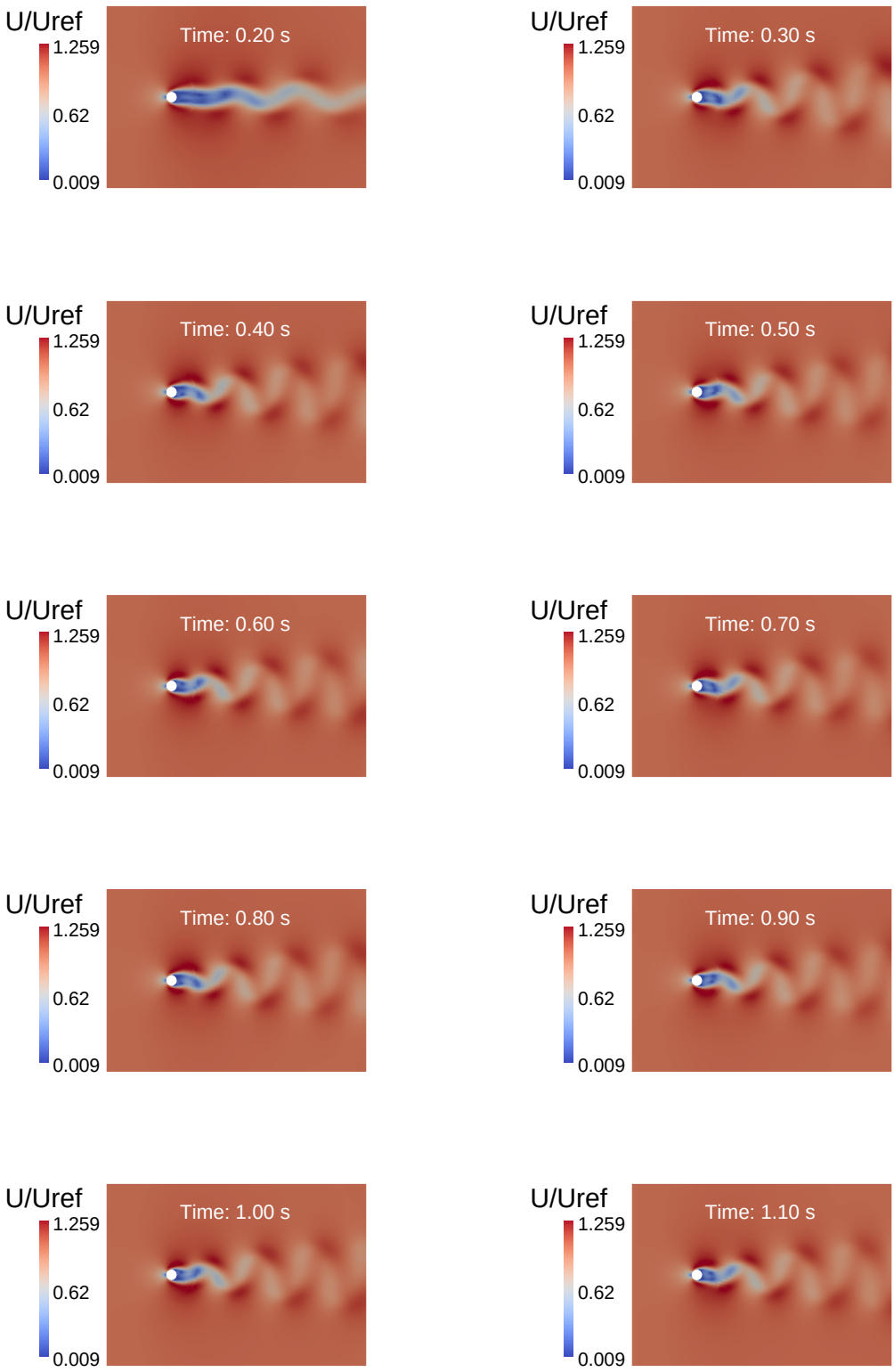

Figure 6.12: VIV evolution of the velocity magnitude field, $R_e = 102$. Left to right, top to bottom.

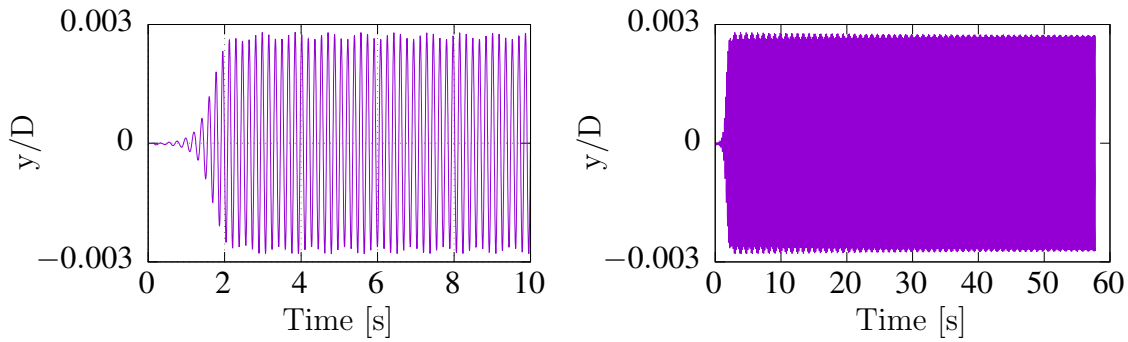

Figure 6.13: Cylinder displacement $R_e = 94$

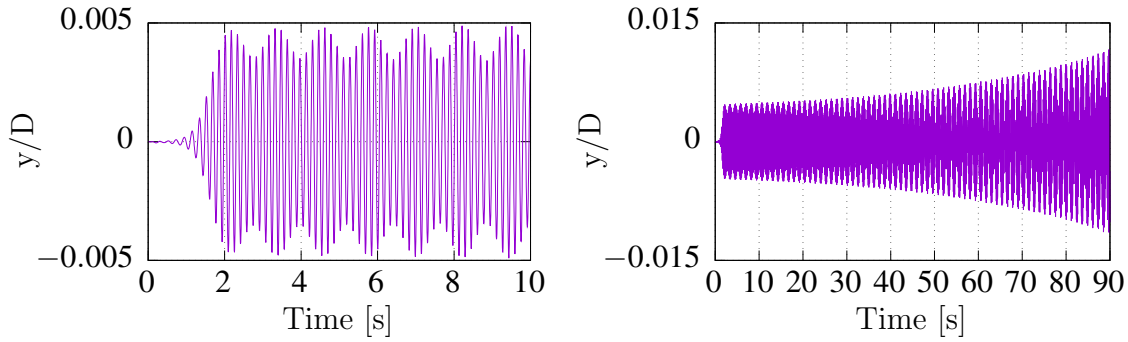

Figure 6.14: Cylinder displacement $R_e = 98$

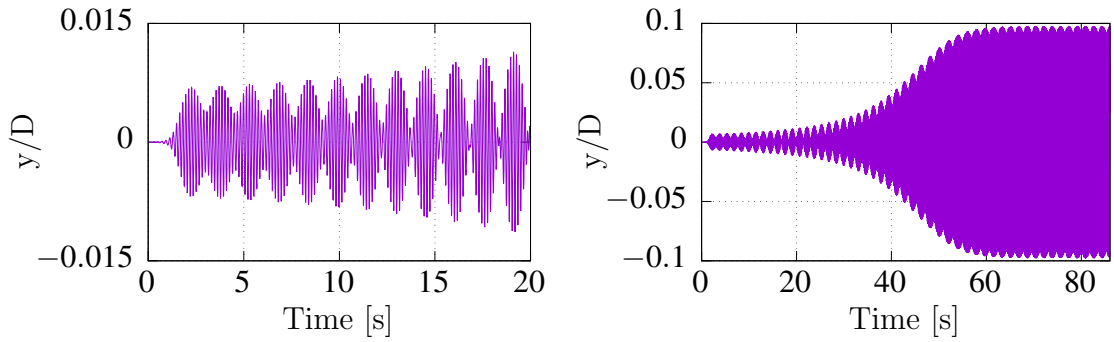

Figure 6.15: Cylinder displacement $R_e = 100$

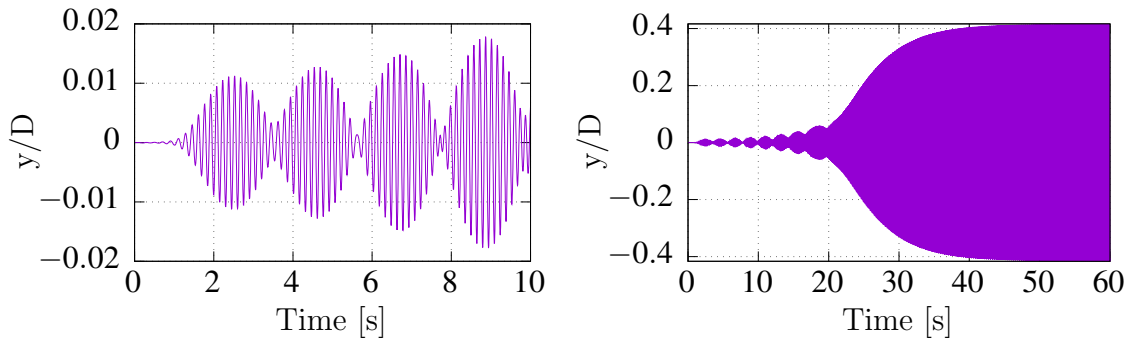

Figure 6.16: Cylinder displacement $R_e = 102$

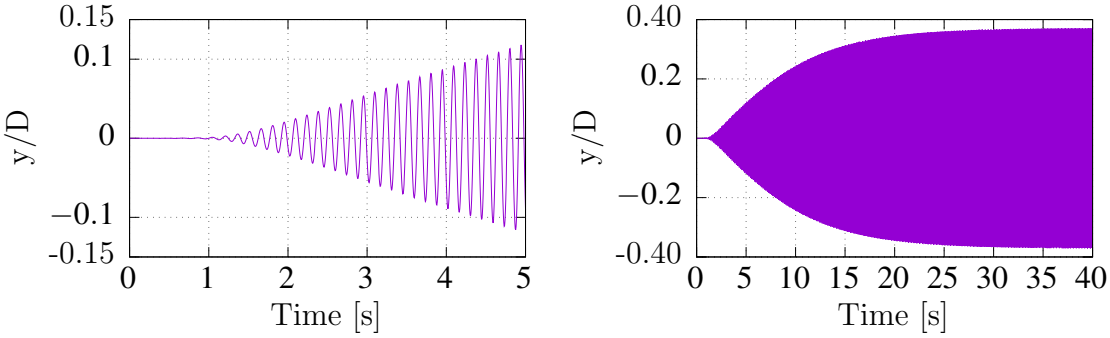

Figure 6.17: Cylinder displacement $R_e = 108$

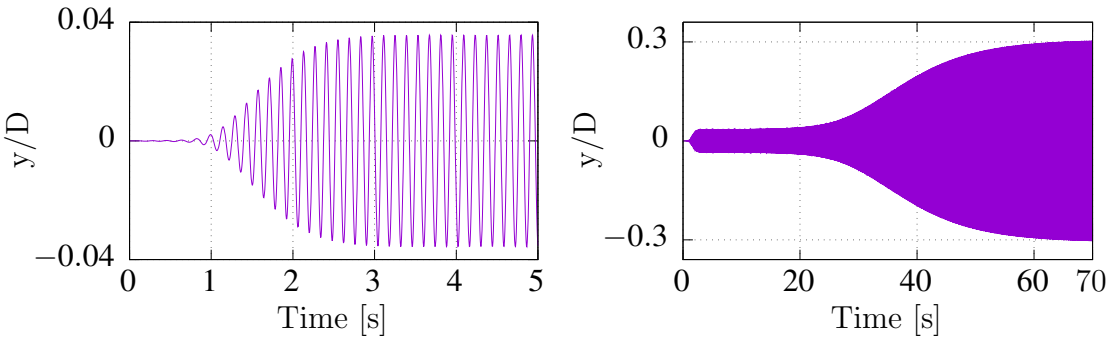

Figure 6.18: Cylinder displacement $R_e = 114$

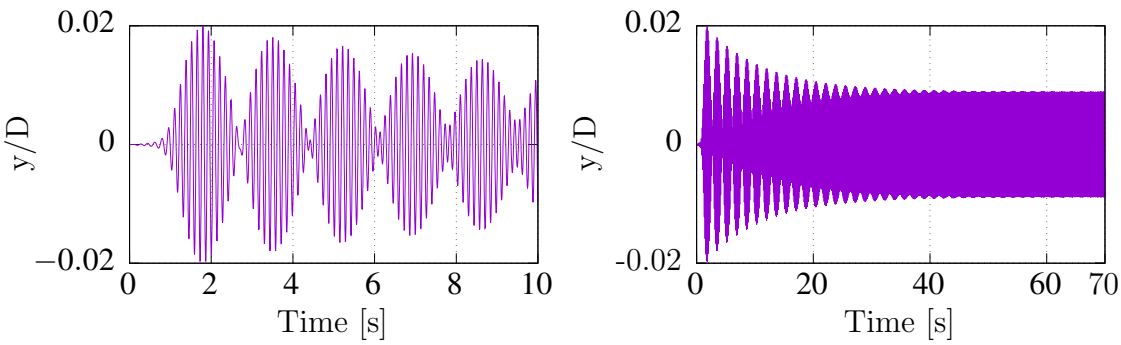

Figure 6.19: Cylinder displacement $R_e = 118$

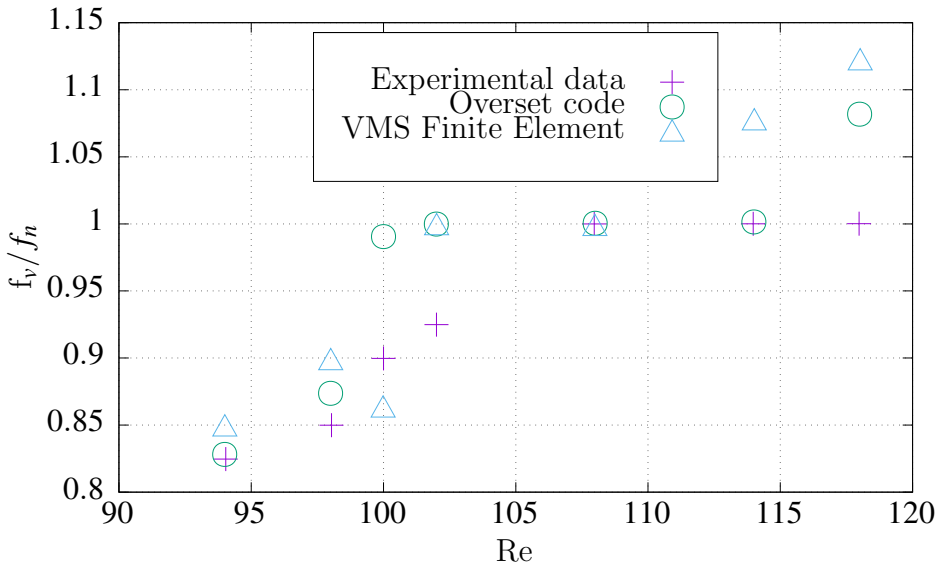

Figure 6.20: Lock-in region frequency comparison.

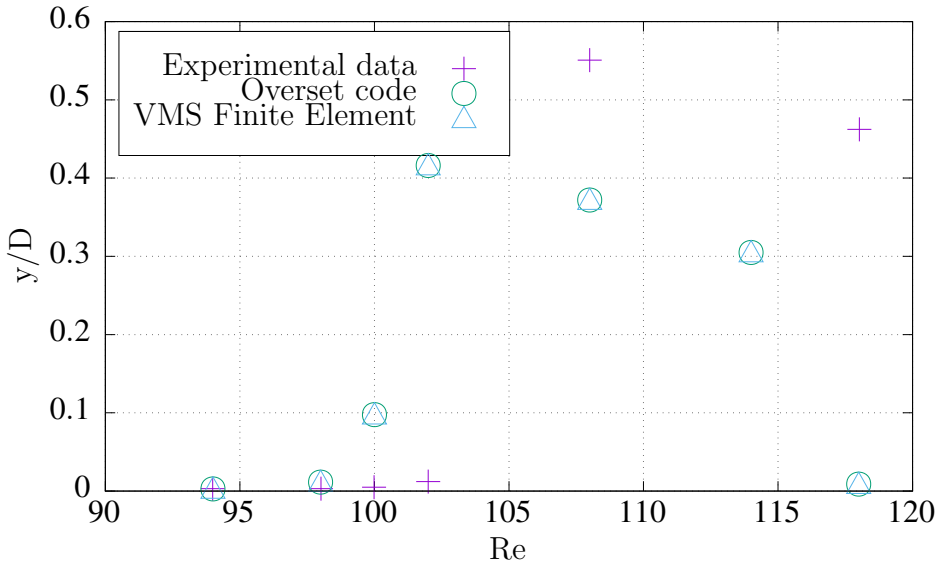

Figure 6.21: Lock-in region final amplitude comparison.

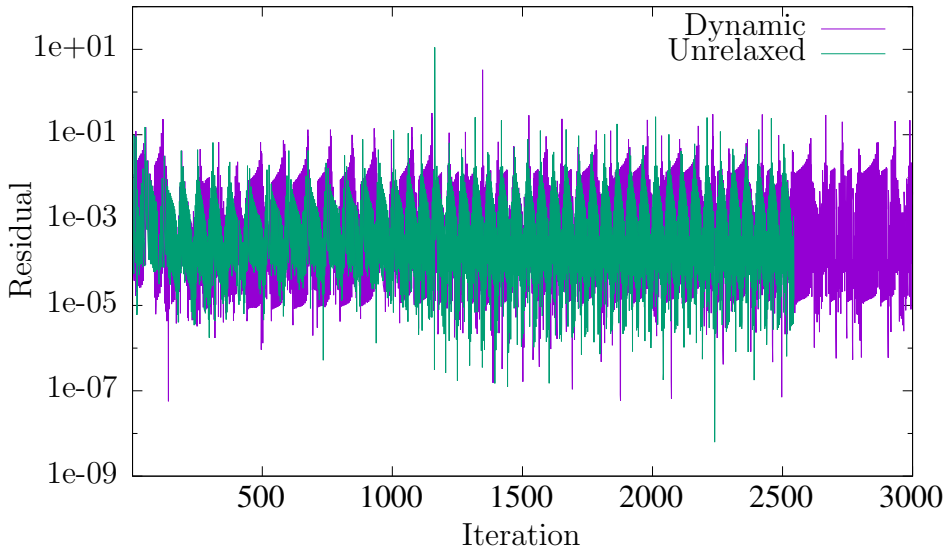

Figure 6.22: Y-momentum Fsi residuals. Bounded Aitkins factors $R_e = 108$.

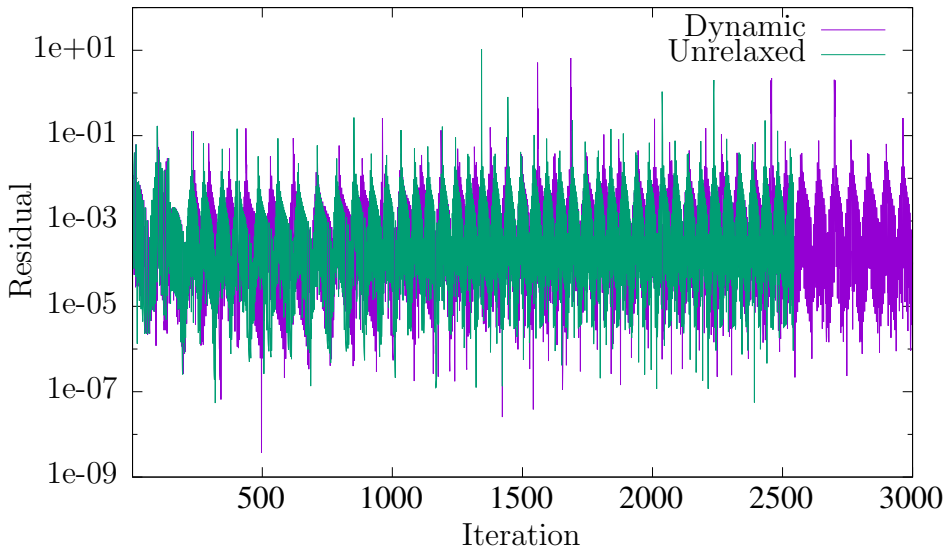

Figure 6.23: Displacement Fsi residuals. Bounded Aitkins factors $R_e = 108$.

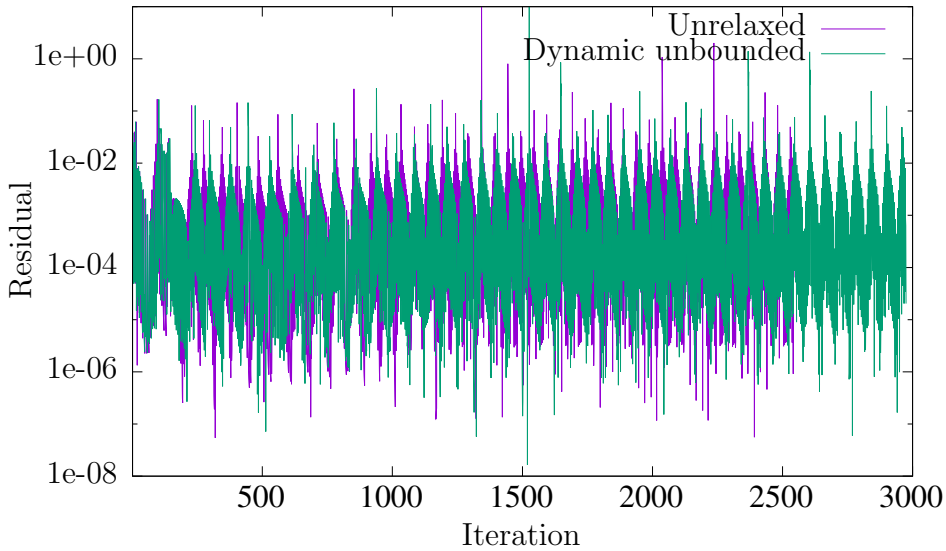

Figure 6.24: Displacement Fsi residuals. Unbounded Aitkins factors. $R_e = 108$.

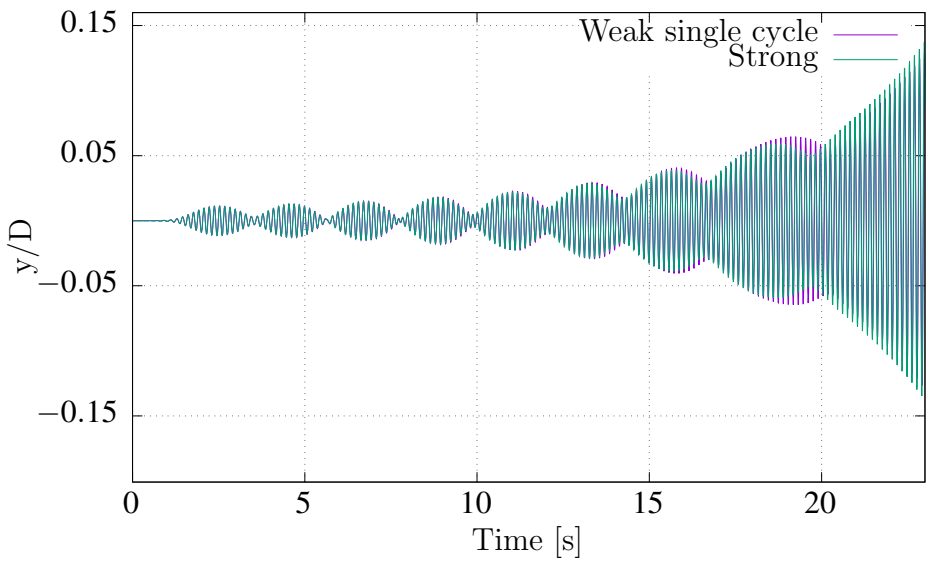

Figure 6.25: Weak (single cycle) vs Strong coupling. Displacement $R_e = 102$.

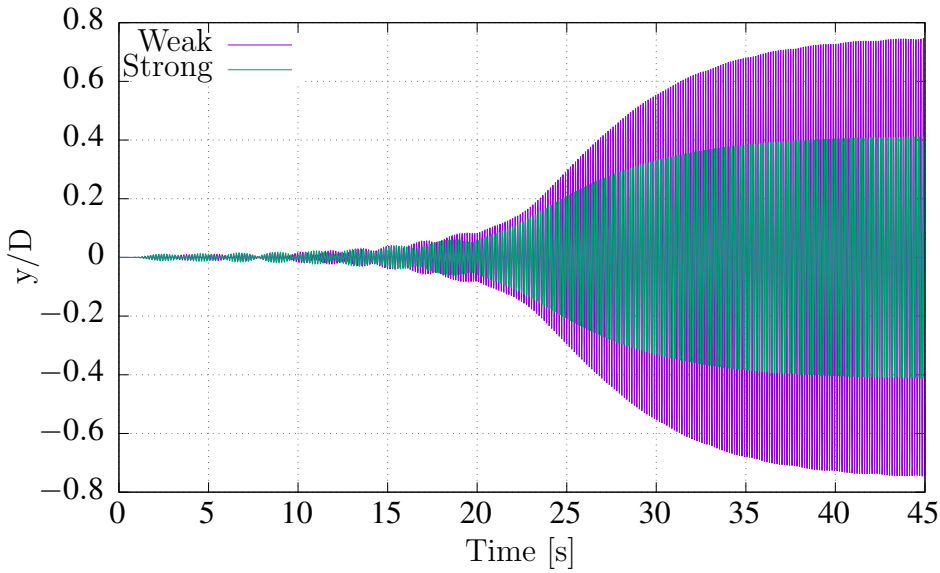

Figure 6.26: Weak (predictor) vs Strong coupling. Displacement $R_e = 102$.

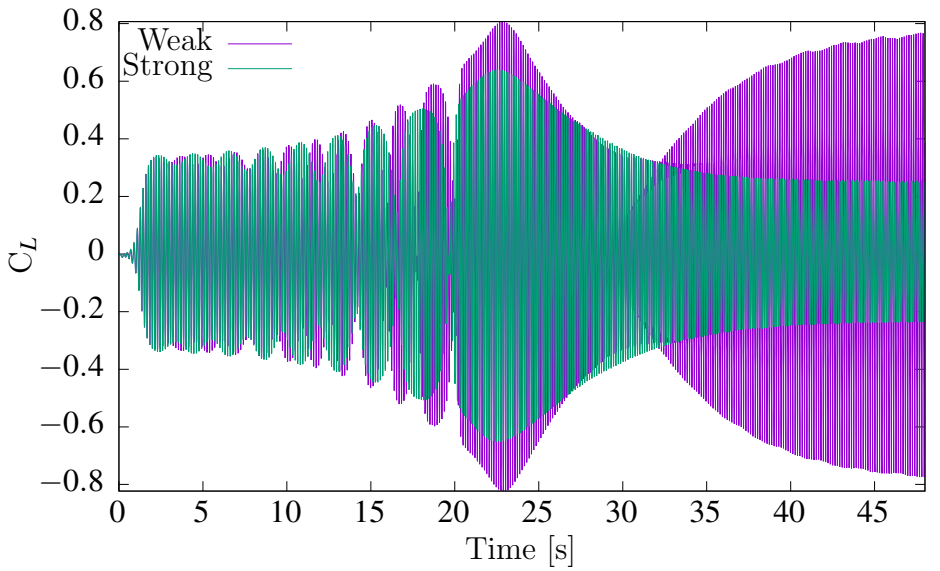

Figure 6.27: Lift coefficient coupling comparison. $R_e = 102$

6.4 Fluttering motion of Rectangular Plates.

In this section the two-way FSI of rectangular plates immersed in a fluid, and under the action of the gravitational field is analysed. In terms of rigid-body motion this case is the most complex presented in this thesis as the solid body is freely able to move in the horizontal and vertical directions, and to rotate about an axis traversing its centroid in the z direction. Its motion then is similar to that of the last case presented in the previous chapter, but allowing an extra degree of freedom in the horizontal direction.

Given the physical and geometrical properties of the plate its motion due to gravitational and fluid forces is predicted, to obtain a benchmark solution. This solution then serves as the basis for exploring the impact of four key factors on the resulting body response, namely: different fluid-solid mass ratios for the same plate dimensions, varying geometrical aspect ratios for the same solid density, different initial conditions, and different Reynolds numbers. The Reynolds number in this case is based on the ratio of gravitational to viscous interactions, and its computation involves the square root of the Galilei number [89] $R_e = \sqrt{\left(\frac{gH^3}{\nu^2}\right)}$. Furthermore, the implications of varying mass ratios are not just physical but also numerical. This is directly reflected in the stability of the solver as a stronger degree of coupling is necessary for low mass ratios. As such, the solver limitations or modifications required under these circumstances shall also be discussed.

The base configuration comprises a $4m$ per side square cavity discretized with a 100×100 uniform background grid, and a rectangular plate of 50 cm width and 5 cm height yielding an aspect ratio of 10. In order to preserve cell quality around the sharp plate corners it has been discretized with four connected structured blocks totalling 5300 cells. Initially the plate is positioned at the origin of the coordinate system which is near the top right corner of the cavity with a 60° orientation. Figures 6.28, and 6.29 show the corresponding computational domain, and a zoom of the spatial discretization around the rectangular plate.

Dirichlet boundary conditions for the momentum equations are employed on the plate wall, where the velocity of each cell is in general different due to planar motion, and it is computed via the rigid-body solver. Time is discretized with constant steps of $1e-03$ s

in all cases, which is sufficient to keep both advection and grid motion based Courant numbers always below 1. The time-dependent term in the governing equations has been discretized with the second order Backward scheme, using Euler implicit discretization as a starter for the first step. As previously mentioned the plate is released with an initial orientation of 60° and with zero angular and linear velocity. The fluid in the cavity is also at rest initially, and it is only disturbed by plate motion. The Reynolds number is $R_e = \sqrt{\left(\frac{gH^3}{\nu^2}\right)} = 434$ using the plate height as a reference length. The ratio of solid to fluid density is 2.7, and the dimensionless moment of inertia is $I^* = 0.232$. The latter is computed as the ratio between the solid moment of inertia, and the equivalent moment of inertia of a cylinder made up of the corresponding fluid [94].

Reference Solution ($R_e = 434$, aspect ratio 10, mass ratio 2.7)

As the plate is released from rest it accelerates in the vertical direction strongly under the influence of gravity. However, the slanted initial position implies there is also a significant force in the negative horizontal direction mainly as a result of the pressure distribution on the lower left surface of the plate (see first frame of figure Fig. 6.30). This can be confirmed by inspecting the early linear velocity decrease (increase in the negative direction) in Fig. 6.31, which plots the velocity of the centre of gravity in the horizontal direction normalised by an estimate of the vertical terminal velocity given in [94] by $U_\infty = \sqrt{2(\frac{\rho_s}{\rho_f} - 1)Hg}$, as a function of time. The orientation of the plate is also greatly affected by this localised lower left surface pressure, and aided by the upper right surface pressure outweighing the upper left and lower right fluid torque contribution, resulting in a linear decrease of the plate angle as seen in Fig. 6.32.

As the plate rotates in the clockwise direction, towards an horizontal position, the vertical component of the pressure force on the lower surface increases, and it is a major factor to momentarily neutralise the gravitational field force. This is reflected in Fig. 6.33 as the centre of mass downward vertical velocity reaches a maximum, and then steadily decreases to the extent of actually becoming positive. At this point the plate moves upward for a few instants as seen in Fig. 6.34, and at the end of the ascension period it

is oriented almost vertically. Under these conditions the ensuing descent is more pronounced since the pressure force exerted on the plate surface acts almost exclusively in the horizontal direction and poses no significant resistance to the gravitational field. Thus, the second downward acceleration period is marked by a sharper downward velocity. Figure 6.36 illustrates the above described initial descent stage by means of the vorticity field where vortex shedding can be fully appreciated.

The impact of initial conditions

The effect of having slightly different initial conditions is explored by modifying the initial plate orientation for $R_e = 1000$. The differences in terms of initial angle are about 10% $\left(\frac{55\,^\circ}{60\,^\circ}\right)$, and 2% $\left(\frac{59\,^\circ}{60\,^\circ}\right)$.

In general, since the initial plate orientation is less slanted than in the original case for both proposed tests, the initial pressure distribution over the plate surface imposes a higher vertical force component on the body to counter gravitational effects. This is seen in Fig. 6.37 where the plate-weight normalised resultant fluid pressure force in the vertical direction is plotted for the three plate orientations. A lower initial angle results in a higher pressure vertical force. As such, the downward vertical velocity of the centre of mass is expected to be lower than for the original case for both test cases. This is confirmed in Fig. 6.38 which depicts the maximum downward velocity attained during the first descent stage. In the same vein, as plate orientation decreases, the horizontal pressure force on both top and bottom surfaces decreases and therefore the horizontal motion presents a lower amplitude for the test cases as seen in Fig. 6.39. Finally the trajectory of the centre of mass throughout its motion range is presented in Fig.6.40. Although there are significant differences in terms of the ending location of the plate, the trajectories exhibit the same features across the board and are not indicative of chaotic motion.

Exploring different solid-fluid mass and aspect ratios.

The solid to fluid mass ratio is one of the fundamental parameters governing two-way FSI. In this section two modifications to the base configuration ensure the impact of different

solid-fluid mass and geometrical aspect ratios is investigated. The most straightforward manner to alter the solid to fluid mass ratio is to vary either the solid or fluid density. Additionally, as mentioned in the literature review section of this thesis, low mass ratios are known to have a significant influence on solver stability. Because of this, it was decided to decrease the solid density progressively in order to assess the numerical stability of this implementation.

For this particular test, two ratios under 1.8 presented considerable difficulties and it was not possible to obtain a solution. Thus, results presented here correspond to a ratio of 2 in contrast to the original ratio of 2.7. The potential stability issue for this kind of problem can be clearly illustrated in Fig. 6.41 by means of an FSI coupling plot, which presents the vertical location of the centre of mass as a function of the number of FSI coupling iterations for the first two time steps. For solid-fluid mass ratios under 1.8 the coupling is so intricate that the partitioned coupling algorithm is not able to stabilize the added-mass effects even when interface under-relaxation was employed. This results in an ever-oscillating kinematic condition for both media which eventually leads the coupling algorithm to diverge. This is due to the extremely sensitive response that both media exhibit upon small variations in their respective forcing conditions. The potentially catastrophic oscillation patterns emerging from unrelaxed solutions are smoothed out by means of relaxation for cases where the solid-fluid mass ratio is greater than 2. Not only does FSI relaxation play a stabilising role but it also accelerates convergence in this case. As seen in the figure, the adjustments in the location of the centre of mass are negligible from the third or fourth corrector stage onward for the relaxed solution, which can lead to significant savings in terms of computation time.

In order to assess the effect of having a different geometrical aspect ratio, the plate width has been increased to obtain a ratio of 17, the reference solution having a ratio of 10. The density ratio, initial conditions, and Reynolds number ($R_e = 434$) remain unchanged. Results for both modifications can be visualised in Fig. 6.42 through the trajectories of the centre of mass during the descent period. It is interesting to note the fact that during the first descent stage the lower mass ratio (2) plate reaches a lower location than its heavier counterpart (2.7). This can be supported by examining Fig. 6.43, which reveals the stock

configuration plate has the most pronounced rotation rate. This implies it faces a much stronger fluid opposition in the vertical direction since fluid pressure forces in the vertical direction become more dominant as the plate approaches a fully horizontal position. The stronger fluid resistance results in the stock configuration plate reaching its first terminal velocity sooner (which by the way is greater in magnitude as one would expect for a heavier object), as seen in Figure 6.44.

The direct consequence of a greater aspect ratio is an increase in the fluttering motion amplitude as seen partially in Fig. 6.42, and Fig. 6.45. In this case the simulation came to an abrupt end before the first cycle was completed, due to insufficient overlap between both grids as the plate progressively approached the left vertical wall.

Reynolds number impact on fluttering motion.

Lastly, the implications of different Reynolds numbers on the first stage of fluttering motion are investigated. The viscosity of the fluid was changed for this set of tests. The Reynolds numbers investigated are $R_e = 1000$, $R_e = 434$, and $R_e = 45$. The mass ratio was set to 2, with an initial plate orientation of 60°.

Figure 6.46 presents the trajectories of the COG of the plates for different Reynolds numbers. Taking as a reference the lower Reynolds number case: decreasing fluid viscosity results in greater horizontal and vertical motion ranges ('Mid' curve in the plot). However, the same trend is not observed as the viscosity is further reduced ('High' curve in the plot). This is supported by Fig. 6.47 which shows the evolution of the plate orientation. By examining this Figure, it is clear the lowest Reynolds number case exhibits the fastest rate of change in plate orientation. Ultimately this means the plate is subject to higher opposing vertical pressure forces sooner than the other cases, resulting in a limited vertical range of motion. This argument also applies for the horizontal motion range; as the plate evolves towards a fully horizontal position faster, the dominant pressure forces in the horizontal direction decrease, preventing the plate from extending its horizontal motion range. Plate orientation for the 'Mid' and 'High' cases is quite similar during the initial phase of the descent inverval. It is only at about halfway of their trajectories that the 'High' Reynolds number case rotates at a faster rate, thus limiting its range of motion

when compared to the intermediate case.

The horizontal fluid force exerted on the plate through the drag coefficient, presented in Fig. 6.48, is higher (less negative) during the initial descent stage for the lowest Reynolds number flow. The same trend can be observed in the lift coefficient with higher peak values for the lowest Reynolds number case in Fig. 6.49. A snapshot of the vorticity field at $t = 1$ $[s]$ as seen in Fig. 6.50 further indicates that in the lower Reynolds number case the plate has already gone past its fully horizontal orientation, where fluid pressure forces have a major impact on the plate vertical motion range.

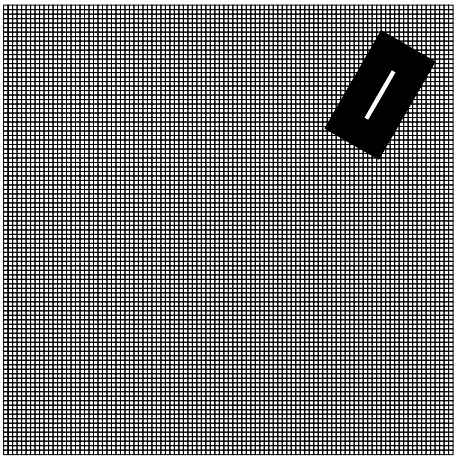

Figure 6.28: Computational domain and initial configuration for the fluttering plate.

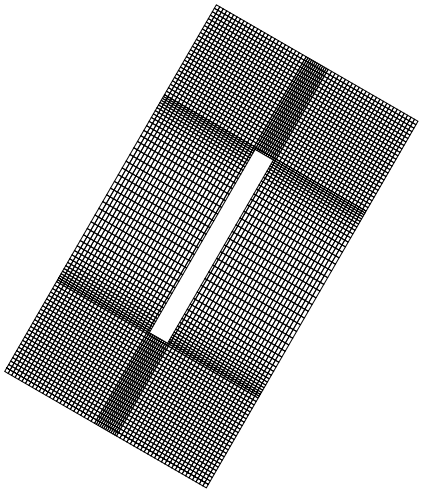

Figure 6.29: Spatial discretization around the rectangular plate.

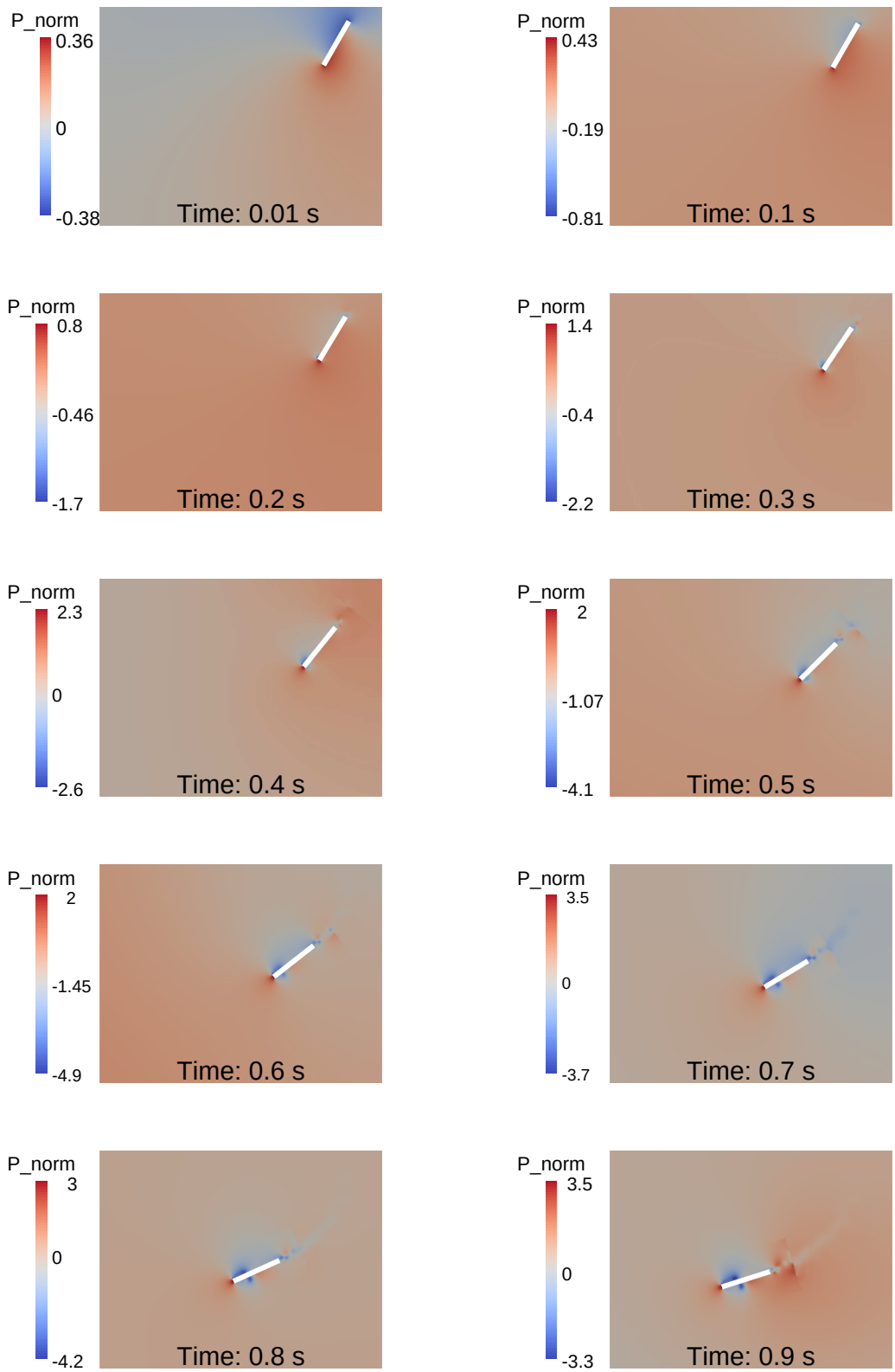

Figure 6.30: Evolution of the pressure field $R_e = 434$. From $t = 0.010\ s$ to $t = 0.90\ s$. Left to right, top to bottom.

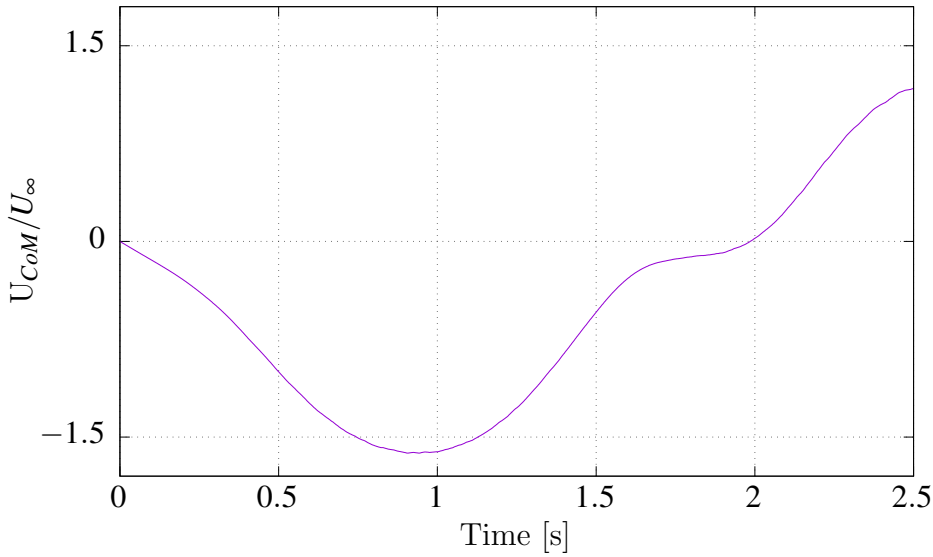

Figure 6.31: Horizontal velocity of the centre of gravity. Mass ratio 2.7.

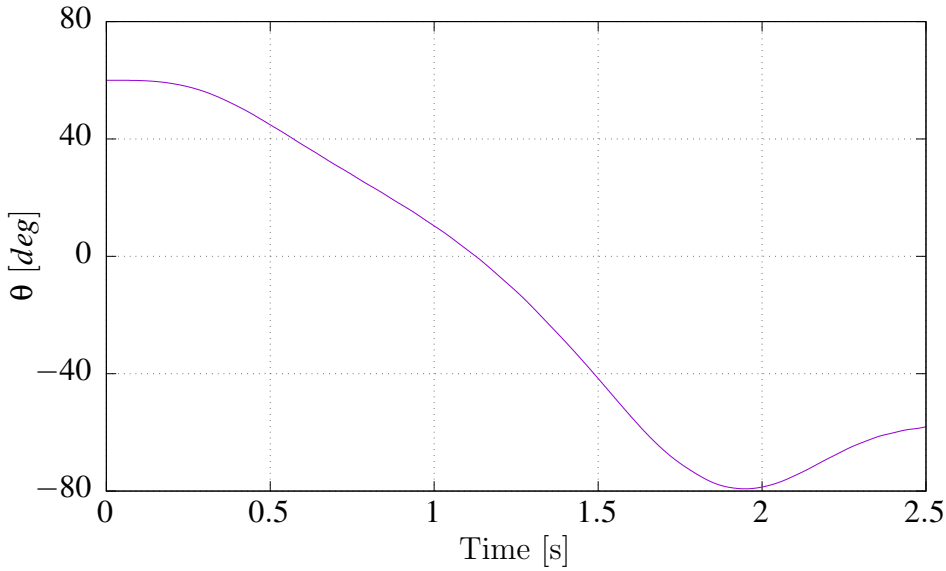

Figure 6.32: Time evolution of plate orientation. Mass ratio 2.7

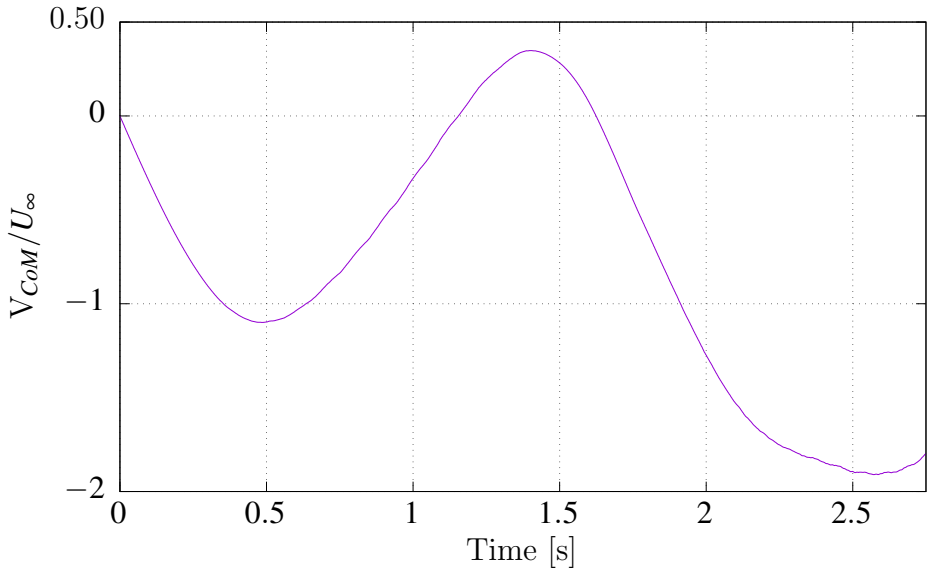

Figure 6.33: Vertical velocity of the centre of gravity. Mass ratio 2.7

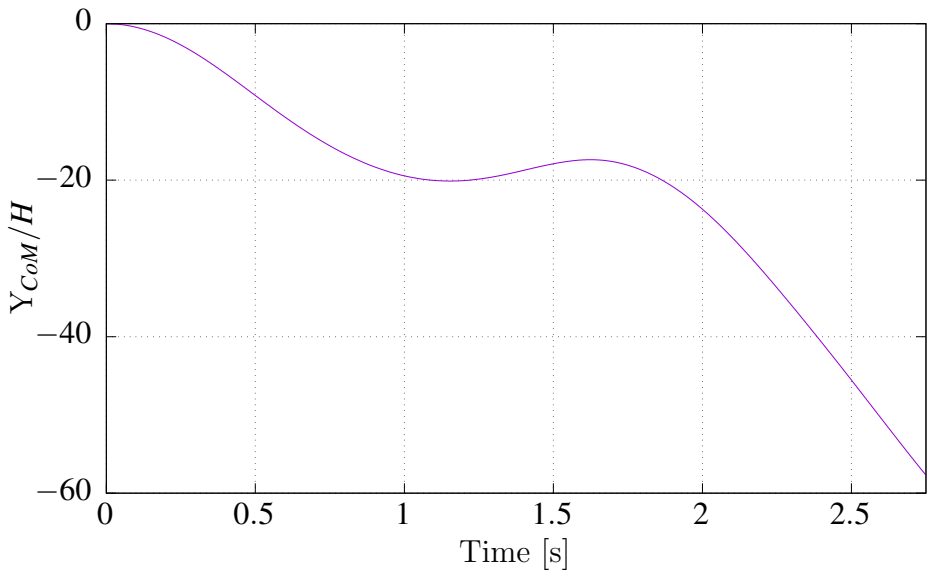

Figure 6.34: Time evolution of the vertical location of the plate centre of mass. Mass ratio 2.7

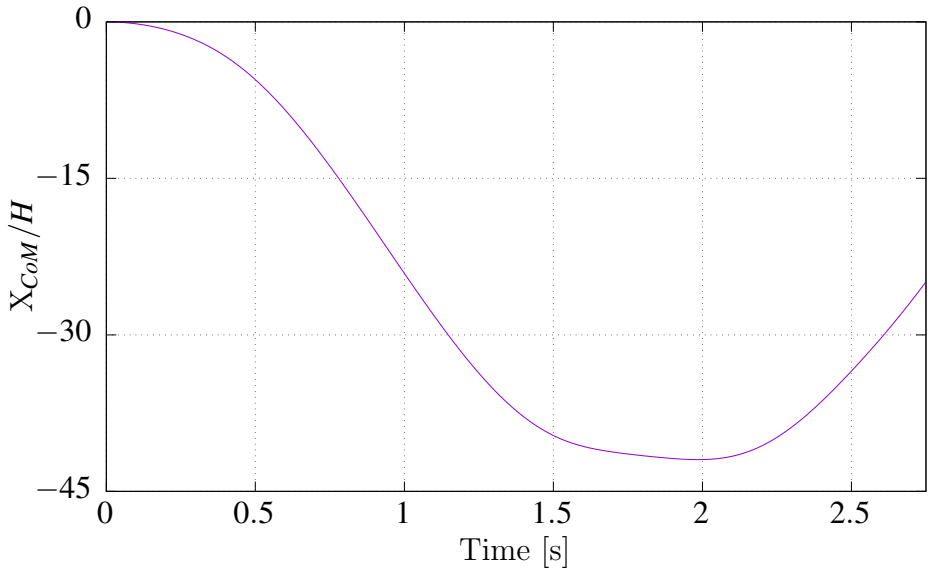

Figure 6.35: Time evolution of the horizontal location of the plate centre of mass. Mass ratio 2.7

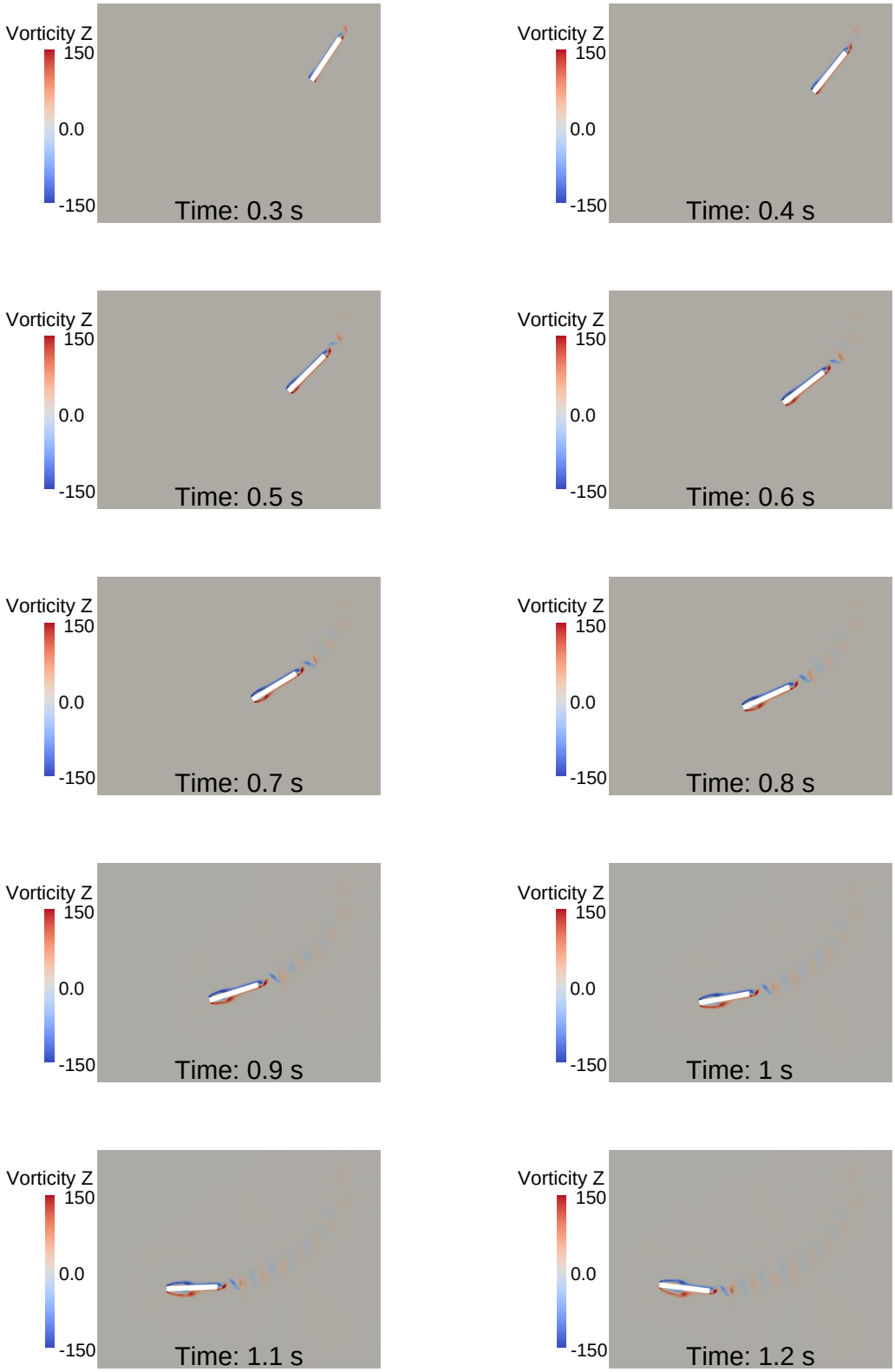

Figure 6.36: Evolution of vorticity field. From $t = 0.30\ s$ to $t = 1.20\ s$. Left to right, top to bottom.

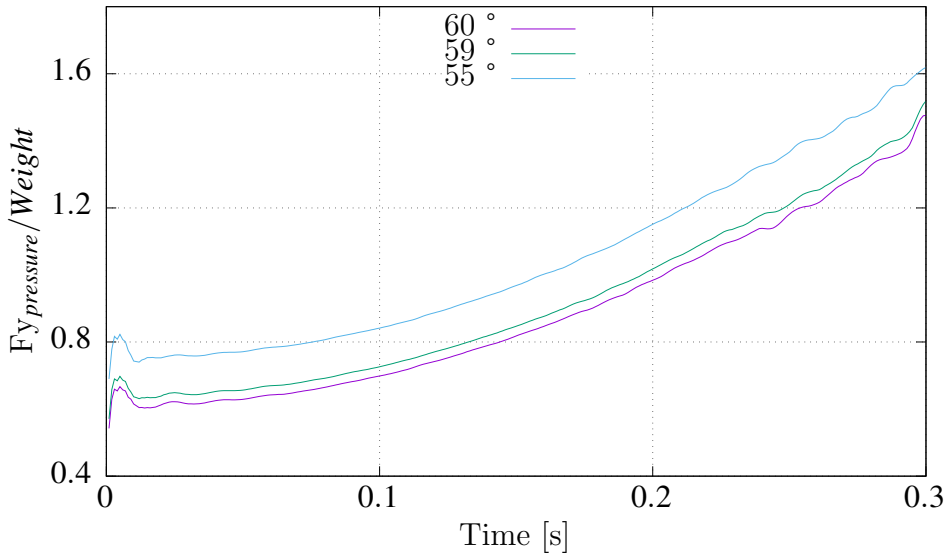

Figure 6.37: Exerted fluid pressure force for different initial plate orientations.

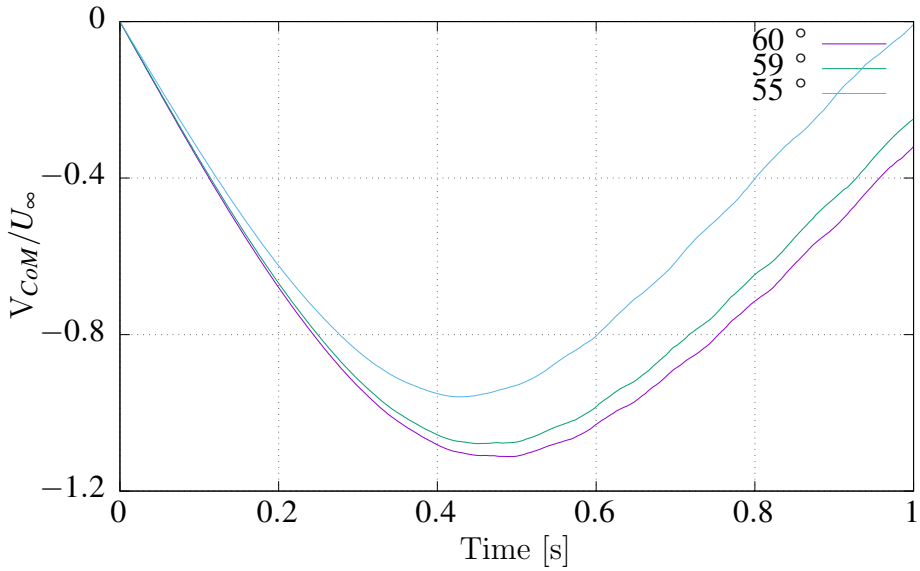

Figure 6.38: Centre of mass vertical velocity for different initial plate orientations.

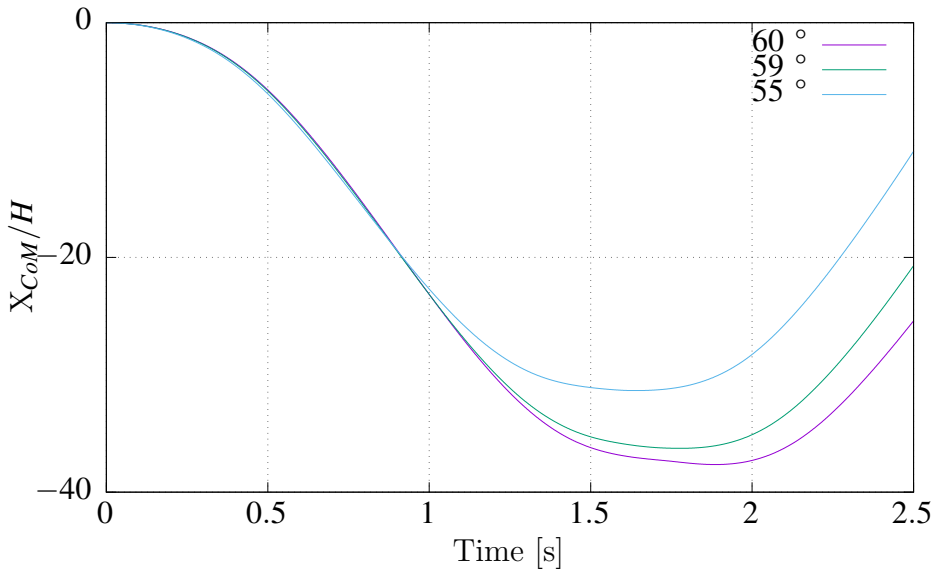

Figure 6.39: Centre of mass horizontal motion range for different initial plate orientations.

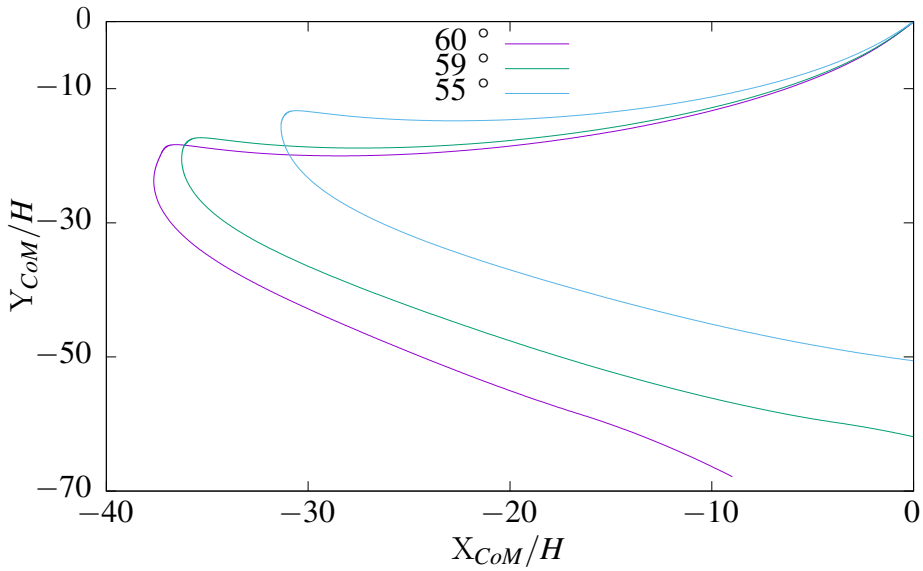

Figure 6.40: Centre of mass trajectory for different initial plate orientations. $R_e = 1000$

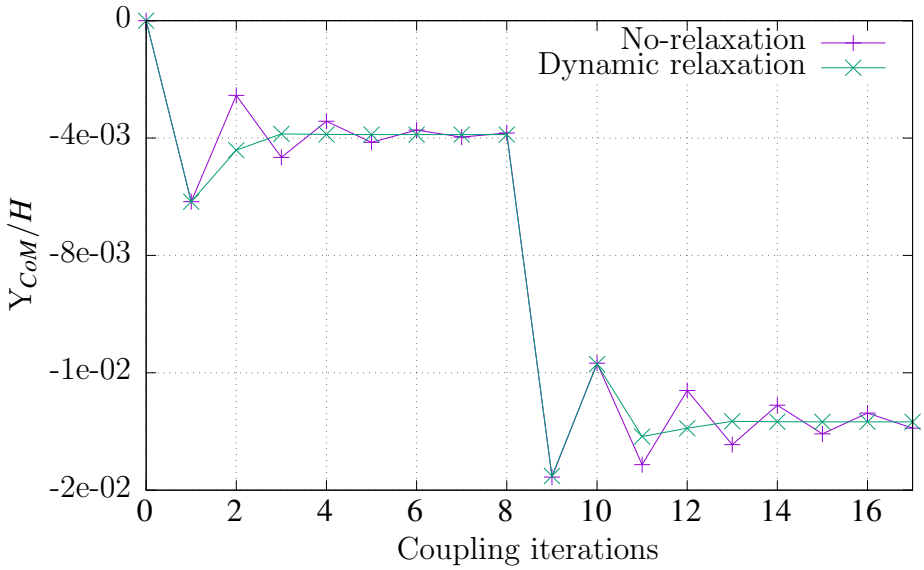

Figure 6.41: FSI Coupling comparison.

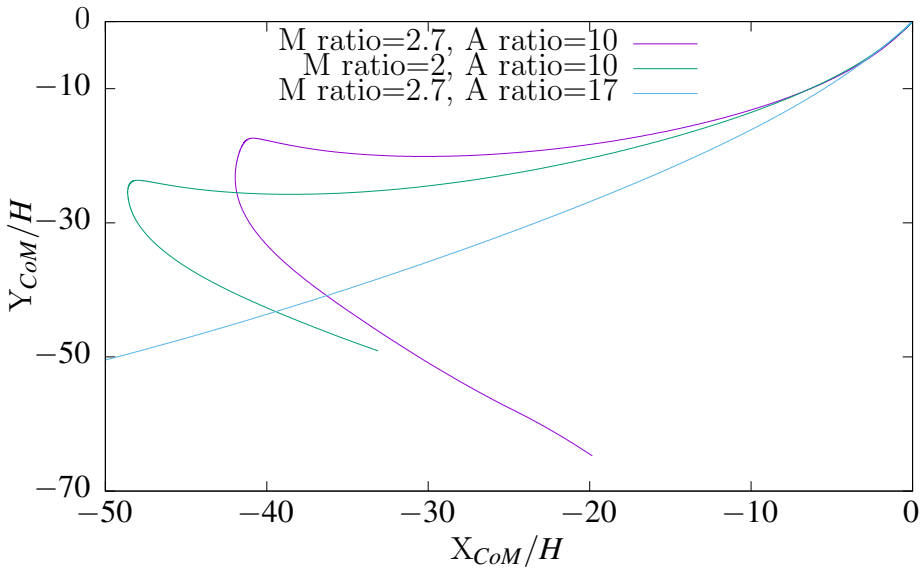

Figure 6.42: Centre of mass trajectory for different geometrical and mass ratios. $R_e = 434$

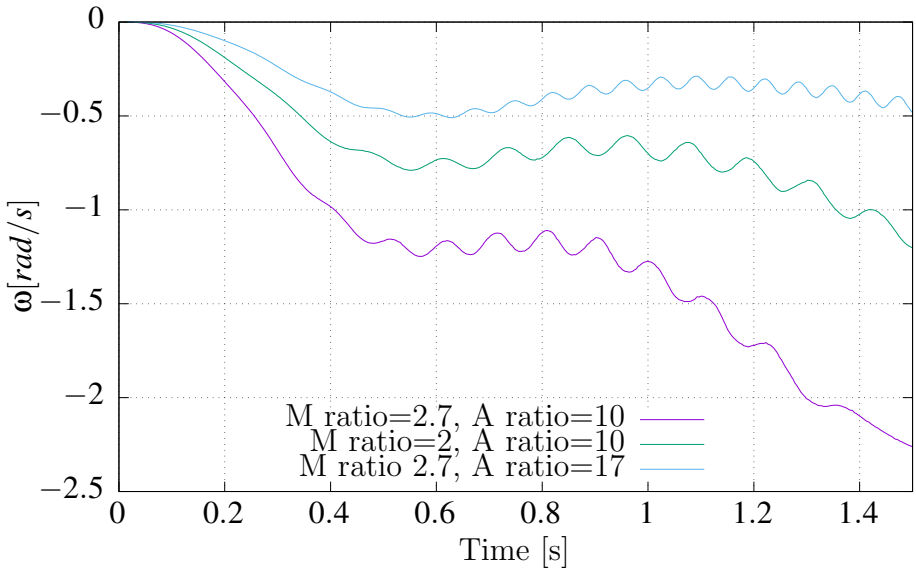

Figure 6.43: Plate ω_z for different geometrical and mass ratios. $R_e = 434$

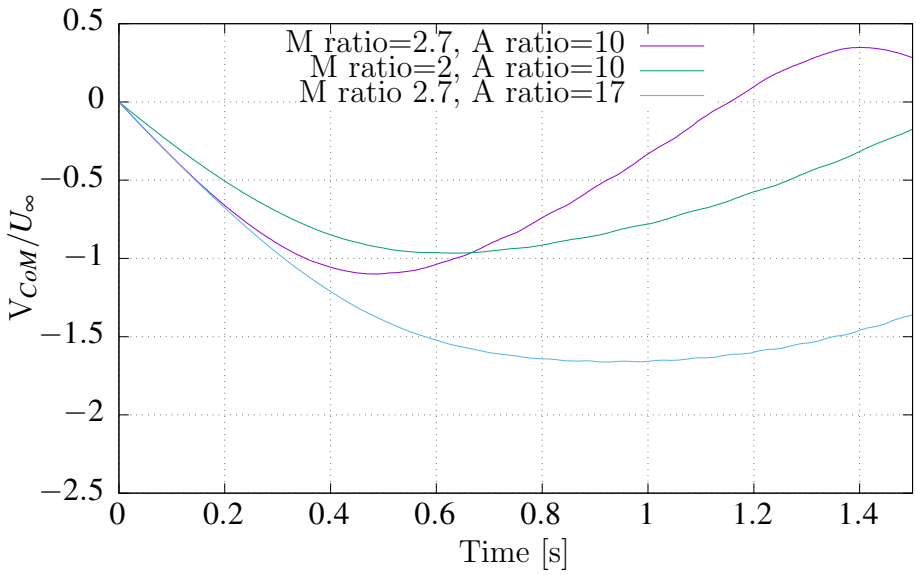

Figure 6.44: Centre of mass vertical velocity for different geometrical and mass ratios. $R_e = 434$

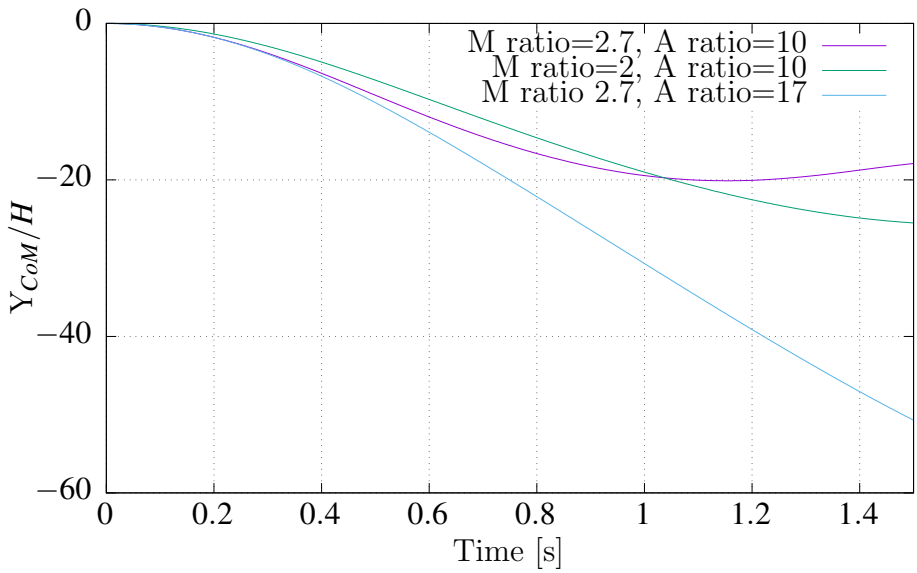

Figure 6.45: Centre of mass vertical location for different geometrical and mass ratios. $R_e = 434$

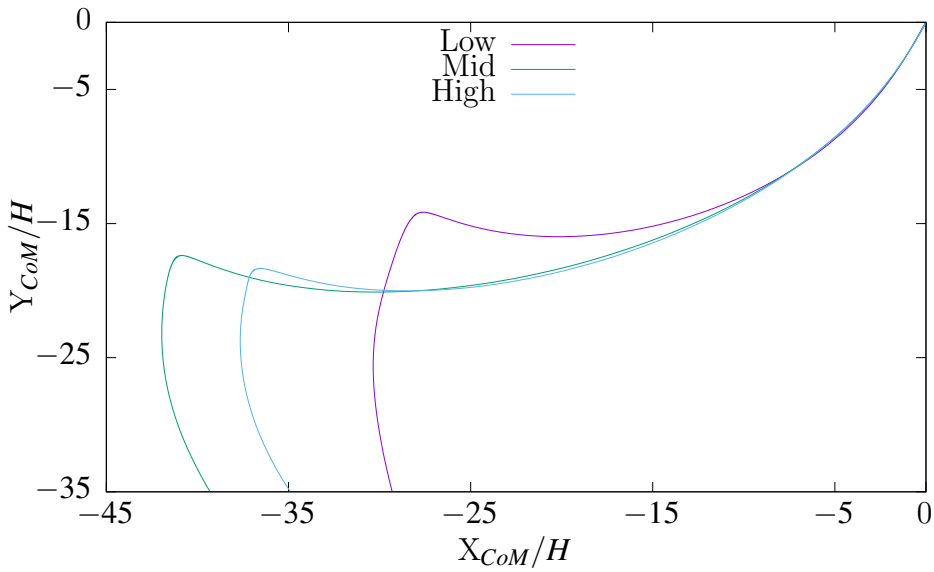

Figure 6.46: Centre of mass trajectory location for different Reynolds numbers. $R_e = 1000$, $R_e = 434$, $R_e = 45$.

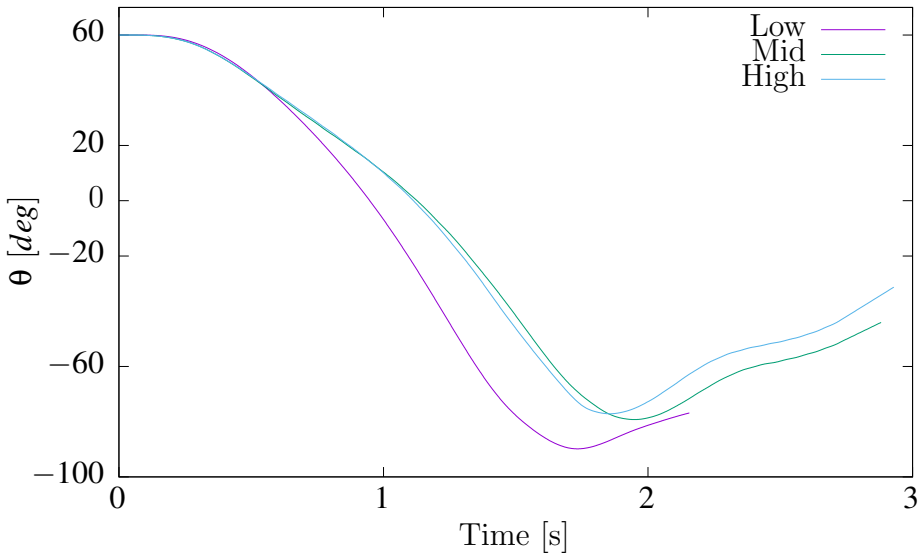

Figure 6.47: Plate orientation for different Reynolds numbers. $R_e = 1000$, $R_e = 434$, $R_e = 45$.

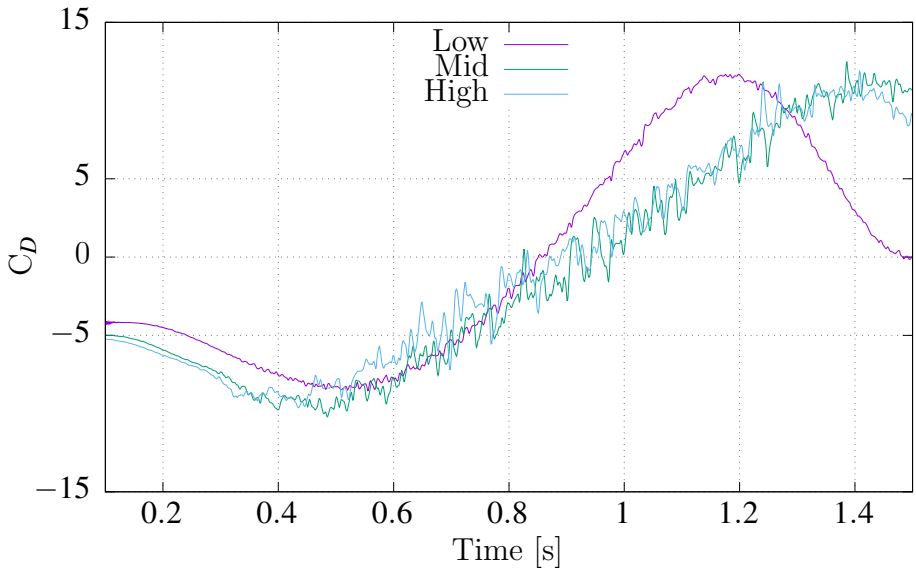

Figure 6.48: Drag coefficient for different Reynolds numbers. $R_e = 1000$, $R_e = 434$, $R_e = 45$.

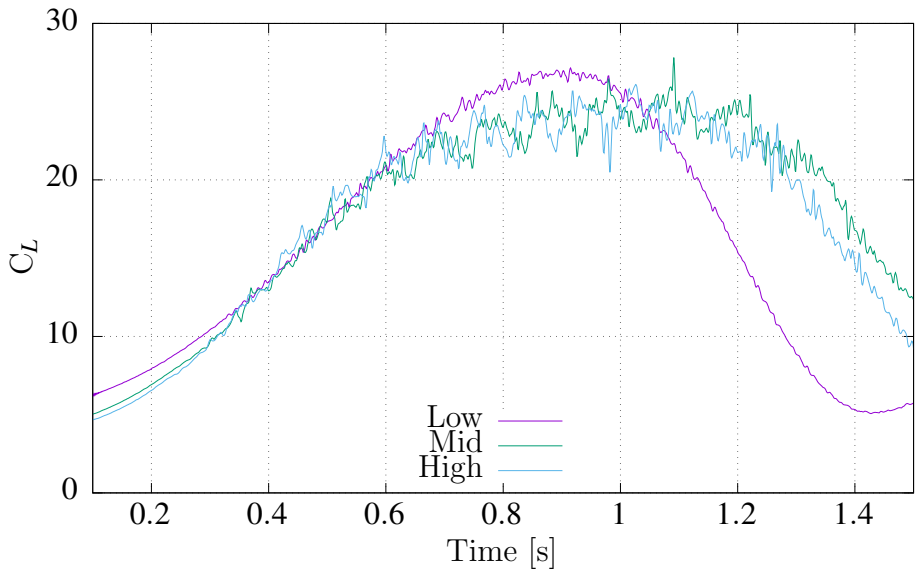

Figure 6.49: Lift coefficient for different Reynolds numbers. $R_e = 1000$, $R_e = 434$, $R_e = 45$.

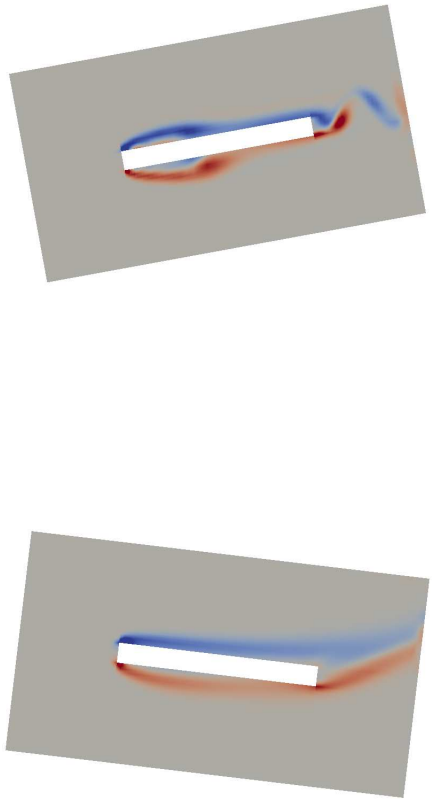

Figure 6.50: Vorticity field for $R_e = 434$ (top), and $R_e = 45$ (bottom) flow regimes. $t = 1s$

6.5 Closure

The two-way FSI coupling algorithm has been validated through two of the three cases presented in this Chapter, and demonstrated its ability to capture satisfactorily highly intricate phenomena such as vortex-induced vibrations. This formally finishes the verification-validation process for all the elements that were incorporated to the original overset solver. Generally speaking, two-way FSI simulations require considerably more computational time than their one-way counterparts due to the iterative nature of the coupling algorithm. In terms of stability, one factor limiting the capability of the current implementation to obtain valid solutions is the solid to fluid mass ratio. Indiscriminate use of interface relaxation is not recommended, and should be evaluated on a case by case basis as it does not always decrease simulation times. If a compromise is required in order to accelerate a given FSI simulation it is advised to at least retain one corrector step for the coupling algorithm, and explore the use of interface over-relaxation.

Chapter 7

Conclusions

This study has presented the extension of capabilities, formal verification, and rigorous validation of a state-of-the art overset mesh methodolody to: the computation of flows which involve moving boundaries with relative motion, the computation of flows generated by multiple moving objects, and the computation of flows which involve two-way rigid-body fluid-solid interaction. This has been achieved through the introduction of several numerical features which, when combined, have resulted in a highly original and effective overset finite-volume rigid-body FSI solver. These include:

- First and second order accurate time discretization schemes for the flow solver.
- General implementation of the Space Conservation Law.
- ALE formulation of the governing equations to handle moving grids.
- Efficient second and third order accurate rigid-body dynamics solver.
- Flexible partitioned two-way FSI coupling algorithm with adaptive relaxation.
- The Method of Manufactured Solutions to formally verify current and future developments.
- The use of overlapping motion tracking grids for two-way FSI problems.

In order to fully exploit the enormous potential of the developed time-dependent overset solver, the space conservation law has been hard-coded to ensure mesh motion does not introduce any non-physical disturbances to the flow field whenever moving grids are

present in a simulation. As seen during the verification process, the current space conservation law implementation allows grid deformation which expands the overset code capabilities. With these contributions in place the Arbitrary Lagrangian Eulerian (ALE) method has been subsequently added to cast the fluid flow governing equations in an inertial reference frame for moving control volumes. Finally, in this first stage of the work, a one-way FSI algorithm has been put in place to enable the modelling of flow around moving objects with prescribed motion.

Another worthwhile and perhaps more interesting branch of the FSI phenomenon is the study of problems for which the movement of solids is not known or prescribed a priori. With the purpose of further extending the overset code feature set, and to simulate this kind of problems, a rigid-body solver has been developed and integrated into a strongly coupled partitioned algorithm.

Additionally, the Method of Manufactured Solutions (MMS) has been incorporated to the source code and has been employed to formally verify the new overset code features. Via a comprehensive set of tests consisting of several combinations, this purely mathematical exercise certified that all the contributions made in this work, along with the base elements developed by Skillen, solve correctly the partial differential equations governing fluid flow. As a by-product of this procedure, the order of accuracy of the method under different circumstances has been computed. Amongst the most interesting results of this phase is the negative impact overlapping grids have on the order of accuracy of the numerical method. This should always be kept in mind when the user needs accurate solutions. Ultimately, it can be considered as a trade-off in exchange for the versatility and high quality grids the method offers compared to non-overlapping approaches to model flow around moving entities.

Whilst the verification procedure certified the core components required for FSI simulations individually, it did not test their combination to solve either one or two-way FSI problems. For this reason, and to provide proof that the proposed fluid and solid models are suitable for laminar incompressible one and two-way FSI simulations, validation has been performed.

The algorithm to solve one-way FSI problems has been validated via three problems

of increasing complexity by comparing solutions with the newly released overset implementation in OpenFOAM®, along with immersed boundary method results for a particular problem. Generally speaking, all solutions compare remarkably well considering the initial grids were not exactly the same for both codes, the existing differences in the overlapping interpolation methods, and the hole cutting algorithms. In terms of maximum computed values for the flow field variables the differences were always around 5%. The last problem presented in the one-way FSI validation process also served to implicitly validate the rigid-body solver for general two-dimensional planar motion.

Similarly, the proposed strongly coupled partitioned two-way FSI algorithm has been validated in the last section of this study. The first case demonstrated very good agreement with a numerical solution using a vorticity based particle solver for the fall of a circular cylinder under gravitational force. This exercise also saw the introduction of a 'motion tracking' intermediate grid as an original contribution. This novel technique allows the user to retain high resolution grids in cases where the motion of the solid object is not prescribed, and grid generation in the overlapping zone is compromised by a disparity in the physical dimensions of the participating entities.

Validation with experimental data and with a Finite Element solution for the case of vortex-induced vibrations of a circular cylinder has also been presented. This case was of particular interest as the overset code was able to fully capture the 'lock-in' resonant phenomenon for the range of Reynolds number considered. Finally, a proposed case study to analyse the impact of different factors such as varying solid-fluid mass and aspect ratios, different flow regimes, and possible chaotic motion via slight variations of the initial conditions on the fluttering motion of a rectangular plate immersed in a fluid has been presented.

The iterative nature of the partitioned two-way coupling algorithm provides the user the flexibility to adjust the degree of coupling as required. The implications on accuracy of different levels of coupling for the two-way partitioned algorithm were exemplified in the vortex-induced vibration case. As discussed, weak coupling is generally not recommended if accurate results are desired. However, a single predictor-corrector cycle could be a feasible option if a compromise between accuracy and computation time is needed,

and the physical fluid-solid coupling is not so intricate.

The application of dynamic interface relaxation via bounded and unbounded variants of Aitken's algorithm showed it is not always beneficial in terms of computation speed-up. Nonetheless, strongly coupled cases such as low solid-fluid mass ratios benefit considerably from its use, and in some cases its omission leads to divergence. The limitations of the current implementation regarding instabilities arising from low solid-fluid mass ratios have been explored. It can be concluded that for ratios below 2 interface relaxation is mandatory, although in some cases even its inclusion does not guarantee converged solutions will be obtained.

Popular alternative techniques to the proposed methodology to solve rigid-body FSI problems have been reviewed, considered, or directly compared through one-way or two-way FSI cases. It can be concluded that within the boundary-conforming-grid family of methods that accurately resolve the fluid-solid interface, the overset-grid approach provides the best compromise, and its use only seems to be limited by the complexity in developing such capability into a flow solver. On the other hand, disperse fluid-solid interface resolving approaches based on the immersed boundary method are far more popular owing to the relative simplicity of the methodology. Both approaches introduce undesirable features into the flow solution such as the loss conservative properties. The overset-grid method introduces this through interpolation whilst the immersed boundary technique does so through the "leakage" of fluid into solid bodies. In the end, if both methodologies are readily available to the user, he/she needs to be aware of the benefits/-drawbacks of both methods to select the most suitable one for the application.

7.1 Future Work

Work in at least three major areas can further expand the versatility offered by the current implementation:

The original work by Skillen included turbulence models. Combining this with the current contributions would open the possibility to explore one-way or two-way FSI turbulent flow scenarios. In hindsight, much (if not all) of the work is already in place since the Reynolds Averaged Navier Stokes approach to turbulence modelling implemented by

Skillen entails the solution of turbulence quantities via transport equations. The code structure laid out by Skillen makes use of a generic transport equation from which each individual fluid flow governing equation stems. Because the time-dependent and advection term contributions made in this work to account for grid motion have been injected into this generic transport equation, it might not be necessary to make any additional changes to account for turbulence modelling. Indeed the good quality of near wall grids, which is a prominent feature of the overset method, makes this the preferred framework for the simulation of turbulent FSI systems. This is because the mean and fluctuating flow fields undergo very strong changes in the near wall regions. In the long term, especially along with the code parallelization recommendation proposed in the next paragraph, it may also be worth considering the implementation of more detailed approaches to turbulence modelling such as Hybrid models (Detached Eddy Simulations, DES) or Large Eddy Simulations (LES) to resolve the large-scale motions arising in the time-dependent boundary-induced flows presented in this work. Both of these techniques would surely provide a more accurate unsteady flow-field representation of the complex interactions between large eddies in contrast to an ensemble-averaged RANS approach providing variations of *mean* conserved quantities as functions of time and space. In the same vein, heat transfer analysis for moving bodies is another appealing possibility. Here again the good quality of the near wall grids offers a distinct advantage in the prediction of the wall heat-flux, which requires the accurate resolution of the wall normal temperature gradient.

The addition of time-dependent modelling capabilities, along with the iterative nature of the two-way FSI coupling algorithm, make code parallelization imperative for the current implementation if it is to serve more practical purposes. Single threaded execution is currently a severe limitation which prevents the use of dense grid computations to obtain accurate solutions. The domain-decomposition essence of the overset-grid method naturally offers a high degree of 'coarse-grained' parallelism [95], i.e. dividing a computer program into large tasks such as assigning individual grids to individual processors for their solution. However, another major process which contributes to the overall computational cost of the methodology is the definition of the computational domain each time step by means of the hole-cutting algorithm. The latter process involves the determination

of interpolation stencils and hole-flagging. The main issue for the efficient parallelization of the overset-grid method with moving bodies is that an optimal partitioning scheme for the parallel computation of the flow field does not provide optimal parallel performance for the determination of the interpolation coefficients (parallelization of the hole-cutting algorithm is challenging on its own, but a useful approach which could serve as a starting point can be found in [95]). Therefore complex load-balancing schemes [95] need to be implemented to improve the overall performance of the parallel implementation. Along the same line, it might be worth exploring the use of more efficient algorithms to accelerate fluid solver convergence such as Algebraic Multigrid solvers.

Finally, as demonstrated in this study, the current space conservation law implementation allows grid deformation, and by doing so enables the possibility for non-rigid fluid-solid interaction modelling using overlapping grids. Depending on the complexity of the deforming structure, a relatively simple ordinary differential equation could be solved numerically to model slender beams under the influence of fluid flow. If the structure demands more detailed modelling, the Finite Volume method could be used to model elasticity as done by Yates[4].

Bibliography

[1] A. Skillen. *The overset grid method, applied to the solution of the incompressible Navier-Stokes equations in two and three spatial dimensions.* PhD thesis, University of Manchester, 2012.

[2] Diane Poirier, Steven R Allmaras, Douglas R McCarthy, Matthew F Smith, and Fancis Y Enomoto. The cgns system. *AIAA Paper*, pages 98–3007, 1998.

[3] H. Jasak and H. G. Weller. Application of the finite volume method and unstructured meshes to linear elasticity. *International Journal for Numerical Methods in Engineering*, 48(2):267–287, 2000.

[4] M. Yates. *Application of the finite-volume method to fluid-structure interaction analysis.* PhD thesis, University of Manchester, 2011.

[5] J. Reddy. *The finite element method in heat transfer and fluid dynamics / J.N. Reddy, D.K. Gartling. Boca Raton, Fla.* London: CRC Press., 1994.

[6] `https://www.nas.nasa.gov/feature_Pegasus_Software_of_the_Year.html`. Accessed: 22/11/2016.

[7] S. Patankar. *Numerical heat transfer and fluid flow / Suhas V. Patankar.* London: Hemisphere., 1980.

[8] J. Anderson. *Computational fluid dynamics : The basics with applications / John D. Anderson.* London: McGraw-Hill, 1995.

[9] J. Ferziger. *Computational methods for fluid dynamics / Joel H. Ferziger, Milovan Perić. (3rd ed.).* London: Springer, 2002.

[10] E. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics A Practical Introduction (3rd ed.).* Dordrecht: Springer., 2009.

[11] Martinović D. Demirdžić, I. and Ivanković A. Numerical simulation of thermal deformation in welded workpiece. *Zavarivanje*, 31:209–219, 1988.

[12] I. Demirdžić and D. Martinović. Finite volume method for thermo-elasto-plastic stress analysis. *Computer Methods in Applied Mechanics and Engineering*, 109(34):331 – 349, 1993.

[13] I. Demirdžić and S. Muzaferija. Finite volume method for stress analysis in complex domains. *International Journal for Numerical Methods in Engineering*, 37(21):3751–3766, 1994.

[14] I. Demirdžić, S. Muzaferija, and Perić M. Benchmark solutions of some structural analysis problems using finite-volume method and multigrid acceleration. *International Journal for Numerical Methods in Engineering*, 40(10):1893–1908, 1997.

[15] Yuan-Ming Wang. Error and extrapolation of a compact {LOD} method for parabolic differential equations. *Journal of Computational and Applied Mathematics*, 235(5):1367 – 1382, 2011.

[16] A.K. Slone, C. Bailey, and M. Cross. Dynamic solid mechanics using finite volume methods. *Applied Mathematical Modelling*, 27(2):69 – 87, 2003.

[17] I. Bijelonja, I. Demirdžić, and S. Muzaferija. A finite volume method for incompressible linear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(4447):6378 – 6390, 2006.

[18] I. Bijelonja, I. Demirdžić, and S. Muzaferija. A finite volume method for large strain analysis of incompressible hyperelastic materials. *International Journal for Numerical Methods in Engineering*, 64(12):1594–1609, 2005.

[19] S.V Patankar and D.B Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15(10):1787 – 1806, 1972.

[20] P. Cardiff, Ž. Tuković, H. Jasak, and A. Ivanković. A block-coupled finite volume methodology for linear elasticity. In 9th *OpenFOAM Workshop University of Zagreb, Croatia.*

[21] P. Cardiff, Ž. Tuković, A. Karač, and A. Ivanković. Nonlinear solid mechanics in OpenFOAM. In 9th *OpenFOAM Workshop University of Zagreb, Croatia.*

[22] P. Cardiff, A. Karač, and A. Ivanković. A large strain finite volume method for orthotropic bodies with general material orientations. *Computer Methods in Applied Mechanics and Engineering*, 268(0):318 – 335, 2014.

[23] P. Cardiff, A. Karač, and A. Ivanković. Development of a finite volume contact solver based on the penalty method. *Computational Materials Science*, 64(0):283 – 284, 2012. Proceedings of the 21st International Workshop on Computational Mechanics of Materials (IWCMM 21).

[24] P. Cardiff. *Development of the Finite Volume Method for Hip Joint Stress Analysis.* PhD thesis, University College Dublin, 2012.

[25] Hedman S. G. An application of the embedded grid technique to the calculation of transonic flow past wings. In *The Aeronautical Research Institute of Sweden, Technical Note AU-1600,Stockholm,1980.*

[26] DS Thompson. A mesh embedding approach for prediction of transonic wing/body/-store flow fields. *Techn. Ber. NASA-CP*, 2201, 1981.

[27] JA Benek, JL Steger, and FC Dougherty. A flexible grid embedding technique with application to the euler equations. 1983.

[28] E.H. Atta and J. Vadyak. A grid interfacing zonal algorithm for three-dimensional transonic flows about aircraft configurations. In E. Krause, editor, *Eighth International Conference on Numerical Methods in Fluid Dynamics*, volume 170 of *Lecture Notes in Physics*, pages 107–114. Springer Berlin Heidelberg, 1982.

[29] Joseph L. Steger and John A. Benek. On the use of composite grid schemes in computational aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 64(13):301 – 320, 1987.

[30] Z.J. Wang. A fully conservative interface algorithm for overlapped grids. *Journal of Computational Physics*, 122(1):96 – 106, 1995.

[31] G Chesshire and W.D Henshaw. Composite overlapping meshes for the solution of partial differential equations. *Journal of Computational Physics*, 90(1):1 – 64, 1990.

[32] Z.J. Wang. A conservative interface algorithm for moving chimera)overlapped) grids. *International Journal of Computational Fluid Dynamics*, 10(3):255–265, 1998.

[33] H.S. Tang, S. Casey Jones, and Fotis Sotiropoulos. An overset-grid method for 3d unsteady incompressible flows. *Journal of Computational Physics*, 191(2):567 – 600, 2003.

[34] Z. J. Wang and Ravishekar Kannan. Overset adaptive cartesian/prism grid method for stationary and moving-boundary flow problems. *AIAA journal*, 45(7):1774–1779, 2007.

[35] Gaofeng Wang, Florent Duchaine, Dimitrios Papadogiannis, Ignacio Duran, Stphane Moreau, and Laurent Y.M. Gicquel. An overset grid method for large eddy simulation of turbomachinery stages. *Journal of Computational Physics*, 274(0):333 – 355, 2014.

[36] Ž. Tuković and H. Jasak. Fvm for Fluid-Structure Interaction with large structural displacements. Course Presentation In *University of Zagreb, Croatia.* 2007.

[37] Matthias Heil, AndrewL. Hazel, and Jonathan Boyle. Solvers for large-displacement fluidstructure interaction problems: segregated versus monolithic approaches. *Computational Mechanics*, 43(1):91–101, 2008.

[38] P.B. Ryzhakov, R. Rossi, S.R. Idelsohn, and E. Oate. A monolithic lagrangian approach for fluidstructure interaction problems. *Computational Mechanics*, 46(6):883–899, 2010.

[39] C. J. Greenshields and H. G. Weller. A unified formulation for continuum mechanics applied to fluidstructure interaction in flexible tubes. *International Journal for Numerical Methods in Engineering*, 64(12):1575–1593, 2005.

[40] C.J. Greenshields, G.P. Venizelos, and A. Ivanković. A fluid-structure model for fast brittle fracture in plastic pipes. *Journal of Fluids and Structures*, 14(2):221 – 234, 2000.

[41] Jean Donea, Antonio Huerta, J-Ph Ponthot, and Antonio Rodríguez-Ferran. Arbitrary lagrangian–eulerian methods. *Encyclopedia of computational mechanics*, 2004.

[42] Emmanuel Lefrancois, Anais Brandely, and Stephane Mottelet. Strongly coupling partitioned scheme for enhanced added mass computation in 2d fluid-structure interaction. *COUPLED SYSTEMS MECHANICS*, 5(3):235–254, 2016.

[43] Christiane Förster, Wolfgang A Wall, and Ekkehard Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer methods in applied mechanics and engineering*, 196(7):1278–1293, 2007.

[44] Charbel Farhat, Michel Lesoinne, and Nathan Maman. Mixed explicit/implicit time integration of coupled aeroelastic problems: Three-field formulation, geometric conservation and distributed solution. *International Journal for Numerical Methods in Fluids*, 21(10):807–835, 1995.

[45] M. Glck, M. Breuer, F. Durst, A. Halfmann, and E. Rank. Computation of fluid-structure interaction on lightweight structures. *Journal of Wind Engineering and Industrial Aerodynamics*, 89(1415):1351 – 1368, 2001. Bluff Body Aerodynamics and Applications.

[46] C. Farhat, M. Lesoinne, and P. Le Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157(12):95 – 114, 1998.

[47] I. Demirdžić and M. Perić. Finite volume method for prediction of fluid flow in arbitrarily shaped domains with moving boundaries. *International Journal for Numerical Methods in Fluids*, 10(7):771–790, 1990.

[48] A.K. Slone, K. Pericleous, C. Bailey, and M. Cross. Dynamic fluid structure interaction using finite volume unstructured mesh procedures. *Computers & Structures*, 80(56):371 – 390, 2002.

[49] J. De Hart, G.W.M. Peters, P.J.G. Schreurs, and F.P.T. Baaijens. A three-dimensional computational analysis of fluid structure interaction in the aortic valve. *Journal of Biomechanics*, 36(1):103 – 112, 2003.

[50] V. Kanyanta, A. Ivankovic, and A. Karac. Validation of a fluid structure interaction numerical model for predicting flow transients in arteries. *Journal of Biomechanics*, 42(11):1705 – 1712, 2009.

[51] Mathieu Olivier, G Dumas, and J Morissette. A fluid-structure interaction solver for nano-air-vehicle flapping wings. 2009.

[52] B.E. Launder and B.I. Sharma. Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Letters in Heat and Mass Transfer*, 1(2):131 – 137, 1974.

[53] H. Bart van der Worp, Leo H. Bonati, and Martin M. Brown. Carotid stenosis. *New England Journal of Medicine*, 369(24):2359–2361, 2013. PMID: 24328480.

[54] Yuichiro Kimura, Hirotoshi Yanagi, Koji Morinishi, Nobuhito Mori, Tomohiro Yasuda, and Hajime Mase. Wave response analysis of flapgate breakwater using fluid-structure interaction model by overset grid method. *Journal of Japan Society of Civil Engineers, Ser. B 2(Coastal Engineering)*, 66(1):811–815, 2010.

[55] Kwang J. P. *Simulation of fluid-structure interaction for surface ships with linear-nonlinear deformations*. PhD thesis, University of Iowa, 2010.

[56] Toshiyuki Nakata and Hao Liu. A fluid structure interaction model of insect flight with flexible wings. *Journal of Computational Physics*, 231(4):1822 – 1847, 2012.

[57] Scott T Miller, RL Campbell, CW Elsworth, JS Pitt, and DA Boger. An overset grid method for fluid-structure interaction. *World Journal of Mechanics*, 2014, 2014.

[58] Joel H Ferziger. Numerical methods for engineering application. Technical report, 1981.

[59] Vuko. Vukcevic. *Numerical Modelling of Coupled Potential and Viscous Flow for Marine Applications*. PhD thesis, University of Zagreb, 2016.

[60] Charles S Peskin. The immersed boundary method. *Acta numerica*, 11:479–517, 2002.

[61] Kunihiko Taira and Tim Colonius. The immersed boundary method: a projection approach. *Journal of Computational Physics*, 225(2):2118–2137, 2007.

[62] Chengjie Wang and Jeff D Eldredge. Strongly coupled dynamics of fluids and rigid-body systems with the immersed boundary projection method. *Journal of Computational Physics*, 295:87–113, 2015.

[63] C.J. Freitas and S.R. Runnels. Simulation of fluid-structure interaction using patched-overset grids. *Journal of Fluids and Structures*, 13(2):191 – 207, 1999.

[64] Nathan C. Prewitt, Davy M. Belk, and Wei Shyy. Parallel computing of overset grids for aerodynamic problems with moving objects. *Progress in Aerospace Sciences*, 36(2):117 – 172, 2000.

[65] H. Hadzic. *Development and Application of a Finite Volume Method for the Computation of Flows Around Moving Bodies on Unstructured, Overlapping Grids.* PhD thesis, Technical University Hamburg-Harburg, 2005.

[66] Pablo M Carrica, Robert V Wilson, Ralph W Noack, and Fred Stern. Ship motions using single-phase level set with dynamic overset grids. *Computers & fluids*, 36(9):1415–1433, 2007.

[67] Zhirong Shen, Decheng Wan, and Pablo M Carrica. Dynamic overset grids in OpenFOAM with application to kcs self-propulsion and maneuvering. *Ocean Engineering*, 108:287–306, 2015.

[68] Hrvoje Jasak. *Error analysis and estimation for the finite volume method with applications to fluid flows*. PhD thesis, Imperial College London (University of London), 1996.

[69] Herman F George and Farrukh Qureshi. Newtons law of viscosity, newtonian and non-newtonian fluids. In *Encyclopedia of Tribology*, pages 2416–2420. Springer, 2013.

[70] Phil Kim. *Rigid body dynamics for beginners: euler angles & quaternions*. CreateSpace, 2014.

[71] Samir Muzaferija. *Adaptive finite volume method for flow prediction using unstructured meshes and multigrid approach*. PhD thesis, Imperial College London (University of London), 1994.

[72] CM Rhie and WL Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA journal*, 21(11):1525–1532, 1983.

[73] K. Mehlhorn. *Algorithms and Data Structures The Basic Toolbox / by Kurt Mehlhorn, Peter Sanders.* London: CRC Press., 2008.

[74] Javier Bonet and Jaime Peraire. An alternating digital tree (adt) algorithm for 3d geometric searching and intersection problems. *International Journal for Numerical Methods in Engineering*, 31(1):1–17, 1991.

[75] I. Demirdžić and M. Perić. Space conservation law in finite volume calculations of fluid flow. *International Journal for Numerical Methods in Fluids*, 8(9):1037–1050, 1988.

[76] V. Pavan. *A Consistent Algorithm For Implementing The Space Conservation Law*. PhD thesis, University of Massachusetts Amherst, 2014.

[77] P. D. Thomas and C. K. Lombard. Geometric Conservation Law and Its Application to Flow Computations on Moving Grids. *Aiaa Journal*, 17:1030–1037, 1979.

[78] F. M. Bos. *Numerical Simulation of Flapping Wing Foil and Wing Aerodynamics.* PhD thesis, TU Delft, 2010.

[79] D. Baraff. *Dynamic Simulation of Non-Penetrating Bodies*. PhD thesis, Cornell University, 1992.

[80] Michiel Weghs. Efficiency Improvement of Strongly Coupled Fluid-Structure Interaction Simulations. Master's thesis, Delft University of Technology, 2010.

[81] Ulrich Küttler and Wolfgang A Wall. Fixed-point fluid–structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43(1):61–72, 2008.

[82] Patrick J Roache. *Verification and validation in computational science and engineering*, volume 895. Hermosa Albuquerque, NM, 1998.

[83] Stanly Steinberg and Patrick J Roache. Symbolic manipulation and computational fluid dynamics. *Journal of Computational Physics*, 57(2):251–284, 1985.

[84] Guerrero. J. *Numerical Simulation of the Unsteady Aerodynamics of Flapping Flight*. PhD thesis, University of Genoa, 2009.

[85] Kambiz Salari and Patrick Knupp. Code verification by the method of manufactured solutions. Technical report, Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US), 2000.

[86] Kintak Raymond Yu, Stéphane Étienne, Alexander Hay, and Dominique Pelletier. Code verification for unsteady 3-d fluid-solid interaction problems. *Theoretical and Computational Fluid Dynamics*, 29(5-6):455, 2015.

[87] F Durst, M Peric, JCF Pereira, and G Scheuerer. Study of laminar, unsteady piston-cylinder flows. *Journal of fluids engineering*, 115:687, 1993.

[88] Sheng Xu and Z Jane Wang. An immersed interface method for simulating the interaction of a fluid with moving boundaries. *Journal of Computational Physics*, 216(2):454–493, 2006.

[89] Jeff D Eldredge. Dynamically coupled fluid–body interactions in vorticity-based numerical simulations. *Journal of Computational Physics*, 227(21):9170–9194, 2008.

[90] Maxim A Olshanskii, Leo G Rebholz, and Abner J Salgado. On well-posedness of a velocity-vorticity formulation of the navier-stokes equations with no-slip boundary conditions. *arXiv preprint arXiv:1708.02241*, 2017.

[91] C.H.K. Williamson, , and R. Govardhan. Vortex-induced vibrations. *Annual Review of Fluid Mechanics*, 36(1):413–455, 2004.

[92] P. Anagnostopoulos and P.W. Bearman. Response characteristics of a vortex-excited cylinder at low reynolds numbers. *Journal of Fluids and Structures*, 6(1):39 – 50, 1992.

[93] C. Samaniego, G. Houzeaux, E. Samaniego, and M. Vzquez. Parallel embedded boundary methods for fluid and rigid-body interaction. *Computer Methods in Applied Mechanics and Engineering*, 290(Supplement C):387 – 419, 2015.

[94] A Andersen, U Pesavento, and Z Jane Wang. Unsteady aerodynamics of fluttering and tumbling plates. *Journal of Fluid Mechanics*, 541:65–90, 2005.

[95] Andrew M Wissink and Robert L Meakin. On parallel implementations of dynamic overset grid methods. In *Supercomputing, ACM/IEEE 1997 Conference*, pages 15–15. IEEE, 1997.