Efficient numerical solution for fluid-structure interaction problems

Gregory Benjamin Walton

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy and Master of Science

University of Leeds

EPSRC Centre for Doctoral Training in Future Fluid Dynamics
School of Computing

October 2023

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Acknowledgements

Firstly, a massive thank you to Prof. Peter Jimack and Dr. Mark Walkley for their continual diligent supervision and patience throughout this process. Also, a thank you to Dr. Yongxing Wang; your guidance was invaluable, particularly during the early stages of this PhD. I am grateful to EPSRC and the CDT in Fluid Dynamics at the University of Leeds for their continued financial support.

Thank you to my examiners for their feedback and interesting discussions.

Some personal thank yous; to Imran and Michael for their encouragement and advice. To Joe, Alasdair, and Rose for always providing a much welcomed climbing break at the gym or crag. And to all the members of the CDT, a broad range of characters with whom it was a pleasure to share this experience.

Thank you to my family for their continuous support.

The final word of thanks goes to Cat. Your support throughout the entire process is immeasurable.

Abstract

Fluid-structure interaction (FSI) problems present a difficult challenge when modelling. They are time-dependent and highly non-linear, both in terms of the fluid and solid models. This project aims to explore numerical techniques to solve FSI problems within an arbitrary Lagrangian Eulerian (ALE) framework using the finite element method. By using a conformal meshing approach allows for the deformation of a single mesh throughout the domain, evolving with the fluid-structure interface.

The importance of using an appropriate pressure space approximation to accurately capture the discontinuous pressure at the fluid-structure interface will be demonstrated and described. The stable Taylor-Hood P_2/P_1 element pair, widely used for purely fluid cases, will be compared with the lower order but discontinuous pressure pairing P_2/P_0. These are compared to the $P_2/(P_1 + P_0)$ pair, which enriches the Taylor-Hood discretisation with a piecewise discontinuous constant pressure, on each element. These finite element approximations are tested on two separate two- and three-dimensional test cases, where the extension of the three-dimensional approximation requires additional consideration to be stable.

For large three-dimensional cases, finding solutions of the resulting linear equation systems by direct solvers is severely limited by memory requirements, and thus efficient iterative methods are necessary. Efficient algorithms that employ block-preconditioned Krylov-subspace iterative methods are well known for the fluid flow problem. We consider the extension of these techniques to the discrete FSI problem,

where the monolithic linear system contains contributions from both the solid and fluid models.

A block preconditioner, which uses algebraic multigrid approximation to diagonal velocity components, combined with an approximation to the action of the inverse of the Schur complement gives problem size-independent iteration counts when applied with GMRES.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

When a fluid force acts on a deformable solid surface, it will cause stresses and strains, these will lead the solid to deform to reach a state to minimise these. Fluid structure interactions can be used to drive rigid body motion, often with the deforming fluid exerting a force on a fixed or free rigid structure. In a more complex setting, with both a deforming fluid and a deformable solid, two-way interaction is observed. In both cases, it involves the multiphysical coupling of the behaviour of the fluid and solid. Fluid-structure interaction also often involves a temporal dependency, resulting in steady-state, oscillatory, or random fluctuations.

Fluid-structure interaction (FSI) problems are typically difficult to solve computationally due to their inherent non-linearity and time dependency [19]. Work done to solve analytical cases often assumes a simplified closed-form solution of the governing partial differential equations [19, 210]. Although experimental methods offer another avenue to analyse FSI problems, they can be costly to set up, require specialist instrumentation, and it is generally difficult to record specific data without interfering with the behaviour of the fluid or structure [99]. Therefore, there has been a research focus on computational methods which are robust and accurate, and this leads to the need for efficient approaches for large three-dimensional problems.

This chapter will introduce some typical applications of fluid-structure interactions (FSI) and their importance in engineering design. It will also introduce some of the numerical methods that are used over the duration of this project, eventually leading to motivation to solve these problems more efficiently. To conclude this chapter, an overview of the structure and content of this thesis is presented.

1.1 Fluid-structure interaction applications

The two-way interaction between fluid and solid in engineering design is an important consideration which is often neglected. Its occurrence in the natural environment is ubiquitous. The need to be able to develop and apply numerical models is important, where an efficient and accurate approach can be integrated into fluid (or solid) simulations.

Fluid-structure interaction is observed in a range of problems; a number of overarching categories are outlined below, with further detailed discussion to follow:

- Aerodynamics; wind turbines design [18], aeroelasticity [70, 157, 212], parachute dynamics [115, 178], civil engineering (bridge and building response) [41, 173].
- Biomechanics; blood vessels [15, 16], heart valves [155], vegetation modelling [50, 111], ciliary beating [71, 91].
- Tribology; elastohydrodynamic problems [176], synovial joints [59].
- Geotechnics; soil mechanics [41], sedimentary flow [83, 189, 198].
- Hydrodynamics; floating structures [37], wave energy harvesting [1], fluid sloshing within containers [163].

Aerodynamics engineering often gives some of the more quintessential examples of fluid-structure interactions. Aeroelasticity is the key phenomenon of aerodynamics which concerns itself with fluid-structure interaction, the elastic response of a solid under aerodynamic loading. Here FSI is an integral consideration in the study of structural integrity and in the extension of the service life of an aircraft, where structural oscillation amplitudes occur mainly as a result of flow separation [45]. Aircraft typically operate in a near-transonic regime, where the local shock effect can also contribute to structural dynamics [70]. Flutter is another aerodynamic phenomenon to consider when modelling aircraft, where these uncontrollable vibrations can be particularly destructive [212].

Fluid-structure interactions are of particular interest to the wind turbine design community: large diameter light-weight blades are susceptible to large deflections, which is an important consideration for fatigue-life analysis, and determining the sensitivity of power generation from wind loading. Furthermore, the interaction is complex, involving a periodically rotating structure (the rotor) within an atmospheric boundary layer flow. Therefore, a popular approach is to consider a rotating subdomain that embeds the flexing wind turbine blades and a Eulerian subdomain for the rest of the exterior flow [18, 104]. Although this does present a difficulty in imposed continuity between the two domains given the interface discretisations are incompatible. For offshore wind turbines, further complexity is introduced with wave motion, where the coupling between the solid with airflow and waves can be treated as a FSI problem [37].

Fluid-structure interaction applications to aerodynamics can present some additional considerations, for example, turbulence modelling [178] or compressibility effect (such as shocks) [106]. Parachute modelling is a prevalent application of FSI in the discipline of aerodynamic engineering [106, 115, 178]. Capturing the dynamics of a parachute is a challenging case, since one must consider a highly deformable material, undergoing a large deformation (in the deployment phase), often embedded within a turbulent flow.

Biomechanic systems also provide a wealth of FSI examples in which deformable soft biological matter is in close interaction with a range of fluids (which can include blood, synovia, or air). Fluid-structure interactions of deformable arteries play an important role in haemodynamic metrics of interest, such as wall shear stress and wall tensions, in the application of cerebral aneurysms [17]. Bazilevs et al. [15] uses an isogeometric analysis to model the interaction of blood (albeit assumed to be a Newtonian fluid) with a nonlinear elastic artery. A study by Bazilevs et al. [16] demonstrates the importance of including the flexibility of the artery wall when modelling cerebral aneurysms. Furthermore, in the modelling of bioprosthetic heart valves, the use of deformable solids has a significant effect on accuracy, damping out oscillations in flow rate [105]. Modern numerical FSI techniques are also beginning to see their application in clinical vascular cases [124]. In another example, Fauci and Dillon [71] examines the use of FSI modelling to capture flagellar beating of spermatozoa in the reproductory process.

Modelling the interaction of vegetation with wind has a range of important applications and interests, as described by de Langre [50]. The motivation here lies mainly in the optimisation of biomass cultivation for food, material, and energy production. A dense layer of vegetation, in the form of a canopy, can affect the atmospheric boundary layer and is parameterised using a poroelastic model [63]. In terms of modelling this FSI application, Ikeda et al. [111] uses the immersed boundary approach coupled with a two-dimensional large-eddy simulation turbulence model.

Elastohydrodynamic lubrication (EHL) is typified by two elastic contacts under very high pressure, separated by a lubricating fluid, for example ball bearings or gears. Fluid structure interaction is used to model EHL for lubrication modelling by Singh et al. [176]. Often a finite element approach is applied to solve these contact problems [35, 153]. Synovial joints act mechanically similar to lubricated contacts, where transient EHL has been demonstrated to model ankle joints [59, 137]. The rheologically challenging properties of synovial fluid, which can behave as a non-Newtonian fluid, interact with a biphasic mixture such as articular cartilage [103] (even considered to have a fibrous structure) or with bone tissue (often assumed to be rigid) [145].

From a civil engineering perspective, the fluid-soil-structure interaction has proven to be an important avenue of research, particularly in terms of the structural response to seismic activity [41]. For example, Liaw and Chopra [127] examine the harmonic response of a tower in fluid, specifically water, they consider a one-dimensional beam undergoing deformation using beam theory, to analyse the fundamental modes of vibration. These simple beam models are often used to examine the response of structures to tectonic activity. In contrast to this general beam approach, rigid fibres can be explicitly modelled in an ambient fluid in the modelling of sedimentary flows [189, 198].

Fluid structure interaction is also an important consideration in the design of damping control systems to reduce the wind-induced response of tall buildings

[207]. In a similar sense, the aeroelastic response of the bridge panel can have a catastrophic impact on structural integrity if omitted [173, 184], however, this is an extreme case.

Structures floating on a free surface are also modelled as fluid-structure interactions. Frequently, in these models, the solid is treated as a completely rigid body, and an elastic response of a deformable solid is not considered [9, 122, 132]. The great interest in these types of floating problems is motivated by a need for renewable energy, both directly in terms of wave energy harvesting [1] and also for floating wind turbines [37].

The interaction between a partially filled flexible tank with a sloshing fluid has a range of engineering interest [163]. These can include fuel tanks during transportation, fluids onboard aerospace vehicles, and elevated water towers (subject to earthquakes). The key challenge for these types of problem is the method for treatment of the fluid free-surface, for example, using a volume of fluid (VOF) approach [196] or a marker-and-cell method [161]. Many of these problem-specific investigations only consider rigid solid bodies and no coupling between the solid and the fluid [163]. In the case of modelling sloshing fluid onboard a satellite, the spacecraft is treated as a rigid structure; however, the fluid still interacts with the motion of its orbit [196]. To capture the deformable baffles in a simple three-dimensional sloshing tank, an iterative fluid-solid coupling approach with ALE, coupled with a VOF method, was used to capture the free surface [68].

Clearly, the applications of the FSI are broad and their motivation originates from a range of different requirements. In some cases, the effects of FSI are an important consideration in the design of safe mechanical components throughout their useful life, particularly when evaluating the implications of fatigue [108]. Clearly, there is a need to accurately capture the FSI and resolve it efficiently to inform engineering design. As the fidelity of engineering models improves, the need to include fluid-structure interactions becomes increasingly important. Where analytical solutions are unobtainable and experimental approaches are infeasible, there is a need for accurate and efficient numerical solutions to these problems.

1.2 Background of FSI

When solving FSI problem numerically, they can be broadly categorised according to two main coupling approaches. Monolithic or fully coupled approaches consider a single discrete system of equations for both the fluid and the solid domains. In contrast, a segregated, or partitioned, approach will iteratively solve the fluid and solid equations, which will involve a separate method to couple each domain.

One of the first numerical techniques dedicated to solving fluid-structure interaction is credited to Peskin [155] in 1972, an immersed boundary method for heart modelling. Published around ten years later are two seminal papers [109] and [57] that outline the arbitrary Lagrangian-Eulerian (ALE) finite element method specifically for modelling transient fluid-structure interaction.

In 1998, Farhat et al. [70] couple a computational structural and fluid algorithm in a segregated approach to model aeroelasticity, however they note one of the key limitations of this method is the lack of conservation of momentum and energy. It is not until the early 2000s that the emergence of a fully coupled FSI system arrives [14, 95, 107, 140], with developments in computing power being the critical factor in solving these much larger problems. Toward the latter half of the decade, based on the early work of Peskin [155] three-dimensional immersed boundaries are applied to perform unsteady FSI computations, albeit limited to rigid solid bodies [52, 79]. Subsequently, it is the immersed boundary method that is used to capture a deformable solid in a time-dependent flow, in three dimensions, specifically to capture the mechanics of prosthetic heart valves [90]. The solid is modelled as a linear elastic material, however an eminent limitation of this immersed boundary method at the time is that two-dimensional shells are assembled from one-dimensional fibres. For leaflet-like structures, such as heart valves, this method is appropriate.

Modern papers focus on three-dimensional fluid-structure interactions that undergo large, complex deformations. To model these more complex systems, there is a requirement for more advanced numerical techniques. For example, the paper of Kamensky et al. [116] considers a Lagrangian formulation with an immersed volume as a thin shell structure to model a prosthetic heart valve, with both the solid and fluid domains undergoing deformation. They find significant errors in the approximation of pressure discontinuities across the solid shell. Continuing with the theme of hemodynamics modelling, Wu and Cai [206] use a parallel scalable domain decomposition approach to solve an FSI problem in three dimensions.

Recently, research focus has been on block preconditioners for three-dimensional FSI problems. For example, building a preconditioner based on each physical component of the fully coupled system (where the fluid, solid, and geometry are monolithically coupled into a single system) [56]. Further multilevel methods as preconditioners are popular for unstructured meshes for FSI problems, which are also scalable both in terms of problem size and number of cores to solve with [121].

Although there have been a range of improvements to the FSI methodologies in recent years within academic research, the effect of fluid-structure interaction is often omitted from engineering models, yet, as discussed previously, its presence is prevalent. Even with recent advances in high performance computing and the broad availability of highly parallel software and hardware, there is still a need to develop efficient numerical solvers.

1.3 Motivation for efficient numerical approach to FSI

Discontinuities in pressure are often observed at the interface between media, as is found at the fluid-solid interface. When a continuous pressure space is used in the finite element scheme, a higher local grid resolution is required at the near-interface region. This increases the complexity of the meshing procedure and increases the solution cost due to the increased number of degrees of freedom. The interface pressure is captured more accurately using a discontinuous pressure space. For incompressible fluids, typically a trade-off exists between high mesh resolution or higher polynomial order of the finite element approximation. To better approximate the near-interface pressure jump, without the need for high mesh resolution, special treatment can be applied. This can include using a discontinuous pressure finite element approximation. However, it is the Taylor-Hood element that provides the best convergence properties for a smooth and incompressible fluid.

To solve a discretised system of equations, an efficient algorithm is required to solve linear systems at each time step. For small systems, it is often more efficient to use a direct method. Direct solvers are generally considered robust since they find a precise solution to the system in the absence of rounding errors. However, for larger problems, for example in three dimensions where the number of degrees of freedom can become very large, they can become prohibitively expensive in terms of time and memory requirement [21]. Alternatively, iterative methods, particularly those belonging to the family of *Krylov subspace* methods, can be a preferable approach. With the aim of improving the reliability and performance of these iterative methods, preconditioners lend themselves particularly well to these iterative solvers. For purely fluid problems, affordable, near-optimal block preconditioners have been the focus of much research [24, 66, 67, 78]. For multiphysics problems, such as FSI, the design of a block preconditioner often depends on the configuration of the monolithic formulation [56, 146].

Building on existing work by Wang et al. [201], developing an energy-conserving ALE scheme for FSI, this project aims to improve the efficiency and accuracy of the ALE FSI scheme specifically in three dimensions. The focus of the project is around the performance and accuracy of the solution in relation to the finite element space choice, specifically the pressure FE space. Further, the solution performance can be improved through the use of preconditioning, specifically for this FSI, finite element, ALE framework. Without the use of adaptive mesh refinement or discrete re-meshing the magnitude of displacement of the solid is limited by retaining sufficient quality of the computational grid.

1.3.1 Aims

This thesis will orientate around two key areas of research:

1. Investigate the using of continuous and discontinuous pressure spaces, and the implication of resolving the pressure field has on the accuracy and per-

formance of the solid deflection solution. Specifically performing numerical investigations in two and three dimensions.

2. Develop a computationally efficient block preconditioning approach to be applied to the appropriate finite element pairing with an ALE framework for FSI.

1.4 Thesis structure

Chapter 2 will discuss existing numerical methods in the context of fluid-structure interaction. It will review further the literature on existing numerical methods for FSI, including iterative solvers and preconditioners. Chapter 3 will outline the ALE approach used in the project, with the finite element framework, and discuss the difference between two- and three-dimensional implementation. Chapter 4 will test three different finite element spaces in two dimensions, analysing the difference between using continuous and discontinuous pressure spaces for FSI problems. Similarly, Chapter 5 will extend this to three dimensions. Chapter 6 will discuss designing a block preconditioning approach for this ALE-FE formulation, comparing with a sparse direct solver and an out-of-the-box preconditioned iterative method. Finally, Chapter 7 will summarise the conclusions and future work.

Chapter 2

Numerical methods for FSI

This section reviews existing numerical method approaches for fluid-structure interaction problems. First, the chapter will outline a taxonomy of FSI algorithms, discussing some of the broad categories of numerical FSI methods, which include coupling, discretisation and meshing. Then this chapter will explore literature around existing numerical methods related to solving discrete FSI problems, specifically sparse direct solvers, iterative solvers, and preconditioning techniques.

2.1 Coupling approach

Fluid-structure interaction solution methods can be classified into two key groups according to the approach used to couple the fluid and the structural components; *monolithic methods* and *partitioned methods*.

For this thesis, the following definitions for these categories are used. Monolithic methods tightly couple the fluid and structural dynamics into a single mathematical description for the entire system. This forms a single, fully-coupled, system of equations, which will then be solved using a single algorithm. The interface conditions are implicitly included in the formulation.

In contrast, partitioned methods consider distinct formulations for both the fluid and solid, which are then solved using separate solution algorithms. Coupling between the solid and fluid solvers must be explicitly considered using an additional outer algorithm, generally including the interface conditions.

2.1.1 Monolithic methods

Monolithic methods offer a particularly robust approach to coupling the FSI system. For large deformation and with strong interactions between the fluid and solid, monolithic approaches can exhibit a more stable solution and better convergence properties [14, 107, 139]. However, for time-dependent problems, a large, nonlinear system of equations is required to be solved at each time step, usually using a fixed-point algorithm such as Newton's method [96] or Picard's method [66].

Consequently, compared to partitioned schemes, monolithic approaches tend to be more computationally expensive per time step. However, to compensate for this additional cost, a larger time step is often achievable, given that a monolithic scheme is more stable and accurate in comparison. When numerically comparing the two schemes, they are found to have a similar computational efficiency (measured by the ratio of the inverse of the error to the number of FSI iterations)[140]. It is worth noting that when comparing these two approaches, from an efficiency perspective the preferable approach will depend on the problem size. Although when a direct comparison is made between the two approaches, for the most part monolithic approaches tend to solve in less time relative to a partitioned method given a number of different FSI cases [55].

One of the key drawbacks of a monolithic formulation is their lack of flexibility. It requires specific coupling which is determined by the formulation of choice, which could be designed with a FSI application in mind. This could constrain the choice of numerical solver, leading to sub-optimal efficiency.

As noted above, formulating with this monolithic approach will lead to the generation of a single, large, sparse system of equations. In some cases, this monolithic system can be ill-conditioned and, due to zeros on the diagonal, be indefinite [107]. To overcome these problems, careful consideration must be given to designing a well-preconditioned iterative solver. This is an area where computational improvements to efficiency can be made.

2.1.2 Partitioned methods

The key advantage of using a partitioned approach for FSI problems is that software is able to retain its modularity, where existing solvers are typically well optimised, applied to each constitutive fluid and solid part [55, 135]. It should be noted that partitioned approaches are often applied in commercial packages, where each existing solver is treated as a black box [54, 194].

In addition to having an alternating solution step for each fluid and solid domain, an additional coupling algorithm is required. This will not only have an additional computational burden but may introduce some form of interpolation error where communication of information between domains is required. Typically, partitioned methods are energy increasing and therefore can be unstable [140]. The staggering of the segregated fluid and solid solvers introduces artificial energy at the interface [157]. These partitioned approaches tend to be used for weak interactions between solids and fluids, and risk diverging under strong interactions [107].

In general, partitioned approaches are the less preferable method, separating the solid and fluid formulation usually results in a loss in efficiency and robustness. These numerical difficulties are the result of the widely reported added mass effect [39, 73, 126]. The interaction between the fluid and the structure manifests itself as an additional mass of the fluid in the structural formulation.

2.2 Discretisation approach

Broadly speaking discrete numerical methods for continuum mechanics can be categorised into four main groups, although these are not specific to fluid-structure interactions problem, they are all applicable. The groups are:

- finite difference methods [89],
- finite volume methods [118, 177],
- finite element methods [96, 167], and
- meshless methods (molecular dynamics, smoothed particle hydrodynamics [129], lattice Boltzmann methods [199]).

A summary of each family of discretisation approach will be outlined.

2.2.1 Finite difference method

The finite difference method (FDM) considers the discretisation of the partial differential equation between neighbouring points. The FDM approach lends itself particularly well to an immersed boundary (IB) method for FSI, as it does not require a body-fitted grid, where a finite difference description is applied to the Eulerian fluid [89]. Although simple to implement due to the regularity of the computational grids, the FDM is limited in use for more complex geometries or methods which require the mesh to move as the solid deforms.

2.2.2 Finite volume method

The finite volume method (FVM) aims to balances fluxes between neighbouring cells using surface (or volume) integrals, as a result it is naturally conservative. Unlike the FDM, it can be applied to an arbitrary mesh and therefore is well suited to more complex geometries. The FVM has been coupled with the immersed boundary method in a monolithic approach [118] and used with finite elements for the fluid domain in a segregated approach [177]. Alternatively, some approaches consider a finite volume discretisation of both the solid and the fluid, however, they are still coupled in a partitioned way [134].

2.2.3 Finite element method

The finite element method (FEM) is applicable to a fully unstructured grid, however the complexities lies in the requirement to derive the weak formulations of the governing equations. This is then discretised using localised, element-based approximations. The FEM is very popular for fluid-structure interaction, where fluid and solid can be coupled together in a single formulation, for example, using the arbitrary Lagrangian Eulerian approach [167]. Alternatives using separate existing solvers for the constituents coupled through an additional algorithm are also possible [96].

In finite element analysis, the continuous operators from the partial differential equations are described in a weak formulation and transformed into a discrete problem by defining a set of basis functions. The Galerkin approximation uses a finite-dimensional subspace (of a Hilbert or Banach space) as the basis for the weak solution. The specific choice of the basis function that coincides with the test functions is known as a *Galerkin* method (more specifically, the *Bubnov-Galerkin* method) [20]. A more general approach considers that the trial and test functions do not coincide; this is named the *Petrov-Galerkin* method [66].

These basis functions can either be element-wise continuous or discontinuous, and are usually specific orders of polynomial. Discontinuous basis functions allow for accurate capture of discontinuous features if they are present in the physical problem. A different finite element space can be used for each solution variable (often velocity and pressure for incompressible fluid problems), leading to *mixed-order* finite element formulations [66].

2.2.4 Meshless methods

Meshless methods is a broad term to describe any approach which discretises the spatial domain without the use of a computational mesh (or grid). These approaches tend to predominately use particle-based mechanics. For example, smoothed particle hydrodynamics (SPH) is coupled with other meshless or grid-based methods to capture structural dynamics [129]. Another popular meshless method is the lattice Boltzmann method (LBM) which solves the Boltzmann equations, recovering the macroscopic physical parameters from the microscopic distribution functions by simple arithmetic calculations [42], instead of directly solving the Navier-Stokes equations. LBM is often used with an immersed boundary method as a robust approach to handle FSI, by applying a simple bounce back scheme [199].

2.3 Meshing approach

Numerical fluid-structure interactions can also be categorised using the meshing technique; either using a *non-conforming* or *conforming* mesh method.

Conforming mesh approaches explicitly track the interface between fluid and solid, treating the interface condition as a physical boundary condition [102]. As the solid deforms, the computational mesh must be moved (or re-meshed) as the solution moves forward in time. This family of approaches allows for the use of a single mesh for both constituents, as the interface is captured by the geometry of the fitted mesh.

Non-conforming mesh methods treat the boundary location and the fluid-solid interface as an additional constraint on the model equations. This allows the governing equations to be solved on separate computational grids for the fluid and solid. It is worth noting that a non-conforming, monolithic approach is possible where the fluid and solid equations are formulated in separate domains but solved in a single system.

2.3.1 Non-conforming mesh methods

When applying non-conforming mesh methods, approaches typically consider a case with two meshes. In the simplest case, the fluid and solid domains are discretised separately. This requires adaptive mesh refinement (AMR) to accurately maintain the fluid-solid interface.

As mentioned previously, it is possible to have a non-conforming approach (e.g. two meshes, one for the fluid and one for the solid) and be solved monolithically. An approach that does so is the fictitious domain method (FDM) with distributed Lagrange multiplier (DLM) [82], with the application of modelling sedimentation [83]. The key features of this method is that the fluid equations are enforced inside and outside the solid, and the flow inside the solid is constrained by the additional solid motion using the DLM.

Fully Eulerian approaches have been used for FSI, using a monolithic formulation and interface capturing approach with a single mesh [94, 205]. In this case, the fluid is kept in an Eulerian frame and the Lagrangian solid must be converted to the Eulerian frame. As the solid must be implicitly defined, capturing the discontinuity at the interface becomes a challenge.

Another popular numerical FSI approach is the immersed boundary (IB) method. A fixed Eulerian mesh is used to discretise the whole domain, with the moving boundary of the solid tracked using a Lagrangian grid. The effect of an elastic solid acting on the fluid is transferred by a local forcing term, calculated as the product of a distribution function and the solid stress. Due to the simplicity in generation of simple grids, comprising of one-dimensional fibres representing the boundary of the solid, in two-dimensional problem, and a fixed background fluid mesh which offers a highly robust approach. This can be extended to three-dimensional problems by considering a two-dimensional surface as the solid boundary. A key drawback of the immersed boundary method, shared with two mesh methods, is the error associated with the interpolation of the velocity and the spread of the force between the boundary and the surrounding fluid [142]. In three dimensions there are additional complications due to the surface meshing at the interface.

2.3.2 Conforming mesh methods

One of the key requirements for fitted mesh methods is that the interface between the fluid and solid is explicitly captured. The grid points are located such that they conform to the fluid-solid interface and will move accordingly as the solid deforms, unlike non-conforming methods, which implicitly recover the interface.

A monolithic Eulerian method for the formulations of FSI uses a mesh fitting approach, usually solving for a single field (velocity) [94]. In the case of large solid deflection, the quality of the mesh is maintained by using efficient remeshing tools [94]. This can introduce an additional computational expense and produce interpolation errors between the old and new meshes, at each time step.

A well-known approach to treating discrete interface conditions is Nitche's methods, which using a Lagrange multiplier as a mortaring approach motivated for capturing discontinuities at interfaces. Originally designed for Dirichlet boundary conditions, it has been further extended to handle fluid-structure interfaces [92]. First applied to fitted meshes [92], it has been further extended to be used in unfitted problems for FSI [36].

Arbitrary Lagrangian Eulerian method

A Lagrangian description is preferable for fluids that are constrained to small motions, whereas the Eulerian description is the popular method for cases where a mesh becomes highly distorted [109]. The Arbitrary Lagrangian Eulerian method compromises between both approaches, where the fluid-solid interface is explicitly tracked as the solid deforms, but the surrounding fluid mesh is permitted to deform in order to retain mesh quality. A separate mechanism from the FSI system must be implemented to control the movement of the mesh.

An ALE approach to FSI can also be formulated in a partitioned way, separating solid and fluid solutions [55, 72]. Here, the fluid velocity and the solid displacement are solved in separate systems, with the interface condition imposed in a Dirichlet-Neumann formulation. An efficient solver can be applied to a large fluid system in cases where the solid occupies a relatively small region of the domain.

In contrast, the ALE approach for FSI can be applied in a monolithic formulation [95, 164, 165, 201], coupling the momentum and continuity equations of the solid and fluid in a single system. Often this must also consider a contribution from an ALE advection term. [164] solve a three-dimensional ALE problem using a geometric multigrid method, where the memory required to solve grows only linearly with the size of the problem.

One of the key characteristics of the ALE method is that the mesh does not move completely in either a Lagrangian or a Eulerian frame. For FSI ALE a body-fitted approach is considered, so at the interface the mesh movement is well defined, and the mesh tracks the interface in a Lagrangian way. Elsewhere the control of the mesh can be controlled through a range of techniques. Although this does not need to be physical, it must maintain sufficient mesh quality to provide an accurate solution. In terms of partial differential equations to control the mesh movement, one of the simplest approaches is to use a harmonic [165] or biharmonic equations. Another popular approach is to solve for a linear elastic [201] or non-linear elastic pseudo-solid, separate from the solid present in the FSI system.

More geometric approaches include the use of a torsional spring analogy to control the dynamics of an unstructured three-dimensional mesh [53]. Similarly another approach uses Radial Basis Functions to interpolate the displacement of the structural nodes to the fluid nodes [49], requiring only a solution of a small system (depending on the radius of influence if using compact support).

The ALE method when applied to the FSI problem for this project is outlined in further detail in Chapter 3.

2.4 Efficient numerical methods

When referring to efficiency, this thesis will predominantly consider two parameters. Firstly, efficiency can be described in terms of time to solve, specifically, the focus is the time taken to calculate the unknowns of a system of equations (where the overhead of building and extracting the system is assumed to be fixed and of a lower cost). Second, efficiency in terms of memory and how much computational memory is required to solve the system. In some cases, it is a trade-off between optimising the time to solve and the memory used, then it depends on what suits the user.

It should be noted that throughout this thesis, in general, a bold typeface is used to represent a space of vector-valued functions, where norms and inner products are component-wise, in keeping with the notation of Elman et al. [66].

2.4.1 Sparse direct solvers

Mesh-based discretisation schemes will typically produce very sparse system, where only degrees of freedom interacting with other neighbouring degrees of freedom in the immediate vicinity will be non-zero. Typically, for sparse systems, a direct solve involves a number of steps. Firstly, the system will be ordered in such a way as to reduce fill-in, after which an algorithm will perform a factorisation, either to just a lower triangular matrix for a symmetric system or a lower and upper triangular matrix for asymmetric matrices. Finally, a back-and-forward substitution sweep will be performed using factorisation to calculate the solution.

Sparse direct solvers are classified into two main approaches, frontal and multifrontal methods. Duff [60] further suggests two other categories, “general techniques” and “supernodal algorithms”.

Frontal methods were developed with finite elements in mind, specifically exploiting the fact the discrete system is constructed from the sum of element-wise coefficient sub-matrices. In essence, the Gaussian elimination, for step k of the factorisation,

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - a_{ik}^{(k)} {a_{kk}^{(k)}}^{-1} a_{kj}^{(k)},$$

is performed before all elements have been assembled, given that the terms in the triple product have already been fully summed. Thus, each variable is eliminated as soon as its row and column are fully summed. If elements are ordered from one end of the domain to the other, the active variables form a front that moves through the domain, hence the name frontal method [62].

Multifrontal methods perform partial eliminations on the dense substructures, which consist of groups of elements. The groups of elements in the substructure are linked to other substructures, to form large substructures. These are monitored through element assembly trees. Multifrontal methods have been developed with the interest of symmetric, indefinite sparse systems [61]. This extension to frontal methods allows for several fronts to occur, which lends itself well to parallel computing. Both the popular solution software packages UMFPACK [48] and MUMPS [2] use multifrontal approaches.

The applicability and efficiency of these direct methods depend greatly on the structure of the matrix. To improve the numerical stability and often reduce the memory requirements of a sparse direct solver, pre-processing steps can be taken [156]. One of these approaches consists of *reordering* the matrix in such a way that the resulting resulting factors from the elimination process will remain sparse [66]. Other approaches such as *block-symbolic factorisation* can be used to allocate the data structure of the system before factorisation, analytically computing the block structure of the factors from the reordering step.

2.4.2 Iterative solvers

As the demand for higher-fidelity simulations increases and for three-dimensional simulations, the number of degrees of freedom is ever growing. Iterative solvers typically require less memory and fewer operations than direct methods, however, they compromise on reliability and accuracy, since they practically only converge to a given tolerance [21], and not to machine precision as with direct approaches.

For large, sparse, discretised linear systems, effective iterative solver strategies focus on two families, Krylov subspace methods and multigrid methods [66]. Multigrid methods will be discussed later in relation to preconditioners, with the focus remaining on Krylov subspace methods. The Krylov subspaces are found using the weak Cayley-Hamilton theorem, which states that the inverse of matrix A can be expressed in terms of a linear combination of powers of this matrix. Given the system $Ax = b$ the Krylov subspace is defined as the linear subspace of degree k spanned by the product of b and the matrix A to the powers of k,

$$\mathcal{K}_k(A, b) = \text{span}\,(b, Ab, A^2b, ..., A^{k-1}b).$$

Technical detail on how this relates to iterative solvers is provided in Section 6.1.2.

Some of the most popular Krylov subspace methods include conjugate gradient (CG) [100], biconjugate gradient (BiCG) [75], biconjugate gradient stabilised (BiCGSTAB) [197], minimal residual (MINRES) [152], and generalised minimum residual method (GMRES) [172]. MINRES offers an advantage over the CG method, that is, it does not assume a positive definite matrix, only symmetry. The generalised approach, GMRES, developed by Saad and Schultz [172], does not require a symmetric matrix and is considered the standard iterative approach for non-symmetric systems. For GMRES the current search direction must be orthogonalised against all previous directions, thus the cost grows with each iteration count. This can be mitigated by the introduction of a restarted subspace.

For monolithic methods, in particular, the implication of the fluid-structure model on the structure of the problem does not have any discernible influence on the choice of iterative method. The choice of method is purely derived from the properties of the matrix. The required properties for a given iterative method are outlined in the table below.

	Symmetric	Positive definite	Notes
CG	✓	✓	The differential operator is required to be self-adjoint.
BiCG	×	×	Does not require the differential operator to be self-adjoint.
BiCGSTAB	×	×	Stabilises rounding errors in BiCG.
MINRES	✓	×	Minimises the residual in each subspace.
GMRES	×	×	Requires as many orthogonal bases to be stored as possible.
QMR	×	×	Converges more smoothly than BiCG, and can avoid breakdowns in almost all cases [77].

Table 2.1: Summary of required properties of iterative methods.

Although the CG variants are appropriate for both symmetric and asymmetric systems, there is a constraint on the definiteness. Quasi-minimal residual (QMR) method solves a reduced tridiagonal system in the least squares sense, in a similar approach to GMRES. However, unlike GMRES where the basis of the Krylov subspaces is constructed orthogonally, QMR constructs the basis bi-orthogonally [13].

2.4.3 Preconditioners

The general idea of the preconditioner is to improve the spectral properties of a preconditioned system. Specifically, a preconditioner aims to cluster the eigenvalues of the linear system, which accelerates the convergence of iterative methods. In theory the aim is to convert the linearised system from its original state, $Ax = b$, into a system for which the iterative method will converge faster by multiplying it by a preconditioning matrix, M,

$$M^{-1}Ax = M^{-1}b. \tag{2.1}$$

Preconditioners work particularly well when applied in conjunction with Krylov subspace iterative methods [171], at each iterative step an effective preconditioner will improve the clustering of eigenvalues and thus increase the rate of convergence. The *condition number* is defined as the ratio of the largest to smallest moduli of eigenvalues $\kappa(A) = |\lambda_{max}(A)|/|\lambda_{min}(A)|$. For a positive definite symmetric system, an effective solver is CG, where the error is bound by the condition number [13, Ch. 2.3.1]. However, this is not the case for all systems, since it does not account for the density of eigenvalues without omitting outliers.

Equation (2.1) outlines the application of a left preconditioned system, a system can also be preconditioned through right preconditioning,

$$\begin{aligned} AM^{-1}y &= b, \\ Mx &= y. \end{aligned}$$

Initially, solve for y, and then solve for the solution vector x. There is little difference in the effectiveness of left and right preconditioning [66], however there can be practical benefit when applied to a specific Krylov subspace method.

In order to design an effective preconditioner, the associated cost of the construction and application of the preconditioner M should be less than the total cost due to a reduction in the number of iterations [147]. This is the trade-off when designing an effective preconditioner. Overall the objective of preconditioning should be to achieve the following criteria in the schematic form,

$$\begin{bmatrix} M \text{ set} \\ \text{up cost} \end{bmatrix} + \begin{bmatrix} \text{Single } \tilde{A} \\ \text{iteration cost} \end{bmatrix} \begin{bmatrix} \text{Number of} \\ \tilde{A} \text{ iterations} \end{bmatrix} < \begin{bmatrix} \text{Single } A \\ \text{iteration cost} \end{bmatrix} \begin{bmatrix} \text{Number of} \\ A \text{ iterations} \end{bmatrix} \tag{2.2}$$

where the preconditioned coefficient matrix is $\tilde{A} = M^{-1}A$ [84].

For an optimal preconditioner, there are three desirable requirements in relation to the schematic (2.2). Given the size of the problem is N (number of degrees of freedom) these are,

1. the setup cost of M is order $O(N)$,
2. application cost of the preconditioner is order $O(N)$,
3. and number of iterations to solve is fixed (even as N increases).

This will provide a preconditioned algorithm with a solution cost that scales as $O(N)$. A scalable preconditioning strategy would approach this cost.

Preconditioners can be broadly characterised in three main categories according to how they are derived [21, 22];

- preconditioners methods based on approximate factorisation of the coefficient matrix [21, 170],
- divide-and-conqueror or multilevel preconditioners [80, 125, 164],
- block and approximate Schur complement preconditioners [66, 80, 147].

Some of the most simple preconditioning techniques are orientated around the linear decomposition of the original matrix. A Jacobi preconditioner only considers the diagonal, $M = \operatorname{diag}(A)$, it is very inexpensive to build and invert. Using this method only requires N divisions and multiplications to apply, the cost is $O(N)$ operations.

The Gauss-Seidel preconditioner is another simple approach which uses the lower triangular part of the matrix, such that $M = D - L$. For a symmetric case, it takes the form $M = (D+L)D^{-1}(D+U)$. Here L, D and U are the strict lower triangular, diagonal, and strict upper triangular decomposition of matrix A, respectively. And finally, a variant of the Gauss-Seidel, the successive over-relaxation (SOR) has $M = \omega^{-1}D - L$. Similarly, a symmetric SOR also exists, with

$$M = \frac{\omega}{2-\omega}(\omega^{-1}D - L)D^{-1}(\omega^{-1}D - U).$$

This method can be over-relaxed or under-relaxed according to whether $\omega > 1$ or $\omega < 1$ [85], when $\omega = 1$ SOR becomes Gauss-Seidel. When using this preconditioner, it requires a solution to a triangular system, and therefore the cost grows linearly with the number of non-zeros in the matrix. Decomposing the matrix into the component has a trivial cost. While inverting D scales like $O(N)$ operations.

Incomplete factorisation methods were some of the early approaches used for forming preconditioning. The factors of a sparse system will often become more dense, this is known as fill-in. An incomplete factorisation drops the extra matrix entries through a desired criterion. For example, consider the complete factorisation into a unit lower triangular and upper triangular matrix, $A = LU$. Calculating L and U, for example, by Gaussian elimination, is expensive costing $O(N^3)$ operations for dense matrices, however, applying these methods incurs a cost of just $O(N)$ operations. For sparse matrices, this factorisation will incur a lower cost. Compared to the incomplete factorisation, $A = LU - R$, since the sparsity pattern of the matrix A is S, $R_{ij} = 0 \; \forall \; (ij) \in S$. Simple in concept, a number of extensions have been developed, including the use of level-of-fills and drop tolerance incomplete factorisations. It is worth noting that they do include a number of drawbacks; such as difficulties around parallelisation and a lack of scalability [21].

Multigrid

The main philosophy behind multigrid relies on solving the problem on a hierarchy of coarser grids which should provide an accelerated convergence rate. Multigrid uses two key strategies to build efficient solution methods [33]. Firstly, generating an improved initial guess on subsequently coarser grid solvers is called *nested iteration*. The second uses the idea of using the residual to relax the error called the *correction scheme*. The outline of this scheme is as follows,

1. Approximate solution to $Av = f$ on initial grid Ω^h to obtain v^h.
2. Compute the residual $r = f - Av^h$.
3. Relax on the residual equation $Ae = r$ on a coarser grid Ω^{2h} to approximate error e^{2h}.
4. Finally, correct the solution on Ω^h with the error obtained on Ω^{2h}, $v^h \leftarrow v^h + e^{2h}$.

Beginning at the finest grid level, a smoothing operation is applied to reduce high-frequency errors, before restricting the residual to a coarser grid. The matrix on a finer grid is calculated as two matrix products, $A_{2h} = RA_hP$. Here R is a restriction matrix and P is a prolongation matrix. This is repeated recursively until the coarsest feasible grid is reached. A direct solver can be used on the coarse grid problem to calculate the error, which is interpolated back through the finer grids. The grids on the right of Figure 2.1 are representative of the coarsening levels used in geometric multigrid. Although physical grid levels are not applied in algebraic multigrid, the philosophy behind solving subsequently smaller problems is the same.

Multigrid has the ability to widely adjust the solution “topography” according to the desired performance; varying the number of layers (depth of cycles), number of cycles, type of cycles (V-cycle, W-cycle, or F-cycle), as well as a range of tuning strength parameters. Figure 2.1 shows a schematic of a typical three-layer single V-cycle multigrid.

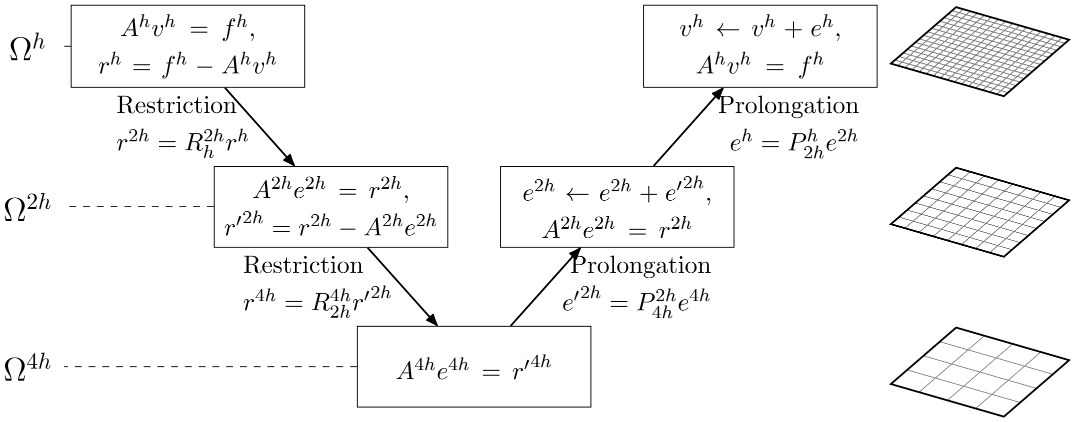


Figure 2.1: A schematic of a single V-cycle of multigrid.

Although originally designed as solvers in their own right, they have also been proven to be effective preconditioners for iterative methods, particularly a robust approach for larger problems. Conventional *geometric multigrid* (GMG) uses a predetermined hierarchy of grids, however, in the case where grid information is not available or is infeasible to access, an alternative is available known as *algebraic multigrid* (AMG). AMG aims to fix the smoother to some relaxation scheme, such as Gauss-Seidel relaxation, and balance efficiency with coarse grid correction by choosing the number of coarsening levels and interpolation appropriately [183].

AMG was developed with a specific class of problems in mind, symmetric positive-definite (SPD) problems, specifically discretised scalar elliptic PDEs of second order. This is not to assume that AMG cannot be applied to a broader range of problems, for example it has successfully been applied to various non-symmetric and indefinite problems [182]. In the case of matrix systems that have large positive off-diagonal entries (a weakly diagonally dominant matrix), an algebraically smooth error can be assumed to oscillate, potentially leading to slow convergence [181].

Although multigrid methods can be used directly as a solution method, they are found to be more efficient than incomplete LU preconditioned Krylov subspace methods for a two-dimensional ALE FSI problem [162]. However, in this thesis the focus will be on using multigrid methods (and AMG in particular) as a preconditioner in combination with Krylov subspace method iterations.

Block preconditioners

In finite element methods for multiphysics problems, including FSI (especially monolithic approaches), it is common practice to exploit the block structure of the discretise a system when applying effective preconditioning. Early research into block preconditioning made use of the saddle point structure of the discrete Stokes [169] and then the Navier-Stokes flow problems [66].

When applying black-box algebraic preconditioners (such as incomplete factorisations or algebraic multigrid methods) to saddle point systems, they often perform poorly. This is due to the indefiniteness and lack of diagonal dominance of these preconditioners, meaning that they are often unstable or of poor quality [24]. For monolithically assembled FSI problems, an efficient preconditioner can be derived from approximations of the different physical components of the problem, for example, considering the solid, fluid and mesh movement components [80]. An advantage of this modular physics-based block approach is the reuse of existing preconditioning methods [112], which can be optimised for each block of the system.

Consider a general purely fluid system. The discretisation of the incompressible Navier-Stokes equations takes a generic block structure of a form

$$\mathcal{A}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{A} & B^T \\ B & -C \end{bmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix}.$$

For these systems typically the C block is zero and so the block structure will take the form,

$$\mathcal{A} = \begin{bmatrix} \boldsymbol{A} & B^T \\ B & 0 \end{bmatrix},$$

for an uniformly stable mixed approximation (which are popular for finite element approaches) of the Navier-Stokes equations [66]. When preconditioning a system as such, a block diagonal structure is often used [24],

$$M = \begin{bmatrix} \boldsymbol{M}_A & 0 \\ 0 & M_S \end{bmatrix},$$

where for the optimal preconditioner the Schur complement operator $M_S = -B\boldsymbol{A}^{-1}B^T$. When using this exact Schur complement and $\boldsymbol{M}_A = \boldsymbol{A}$ a Krylov subspace method will converge in at most three iterations [147]. However, this exact Schur complement requires the inverse of the $\boldsymbol{A}$ block of the original discrete system. When this is computationally infeasible, an appropriate approximation to the Schur complement, M_S, must be used to substitute. For Stokes flow, this block diagonal structure is found to be a very effective preconditioner, therefore the underlying structure is also applied to Navier-Stokes. Similarly for an asymmetric system, the block upper-triangular preconditioner can be utilised,

$$M = \begin{bmatrix} \boldsymbol{M}_A & B^T \\ 0 & -M_S \end{bmatrix},$$

noting the divergence operator in the upper right block and the negative M_S.

When considering a Stokes problem, a good approximation for the Schur complement, M_S, is the pressure mass matrix, Q, or a similar operator (such as the diagonal or lumped pressure mass matrix). The sparse pressure mass matrix, Q, has been shown to be spectrally equivalent to the dense Schur complement, $B\boldsymbol{A}^{-1}B^T$ for Stokes problem [202]. Although this approximation is applicable to discrete Navier-Stokes systems [64], it does not take into account the effect of convection and will lose optimal performance for flows with higher Reynolds number.

One form of approximation of the Schur operator considers a pressure convection-diffusion (PCD) preconditioner,

$$\boldsymbol{M}_S = Q\boldsymbol{F}_p^{-1}\boldsymbol{A}_p,$$

where Q is the a pressure mass matrix, $\boldsymbol{A}_p$ is a sparse pressure Laplacian matrix defined as

$$a_{p,ij} = \int_\Omega \nabla\psi_j \cdot \nabla\psi_i,$$

where ψ is the basis function for the pressure discretisation. $\boldsymbol{F}_p$ is a convection-diffusion operator,

$$f_{p,ij} = \nu\int_\Omega \nabla\psi_j \cdot \nabla\psi_i + \int_\Omega (\boldsymbol{w}_h \cdot \nabla\psi_j)\psi_i.$$

Here $\boldsymbol{w}_h$ is the discrete convection velocity of the nonlinear operator, and ν is the viscosity of the fluid. The assembly of matrix $\boldsymbol{F}_p^{-1}$ presents a particular difficulty. One approach is to consider a single cycle of algebraic multigrid as an approximation to $\boldsymbol{F}_p^{-1}$. A drawback of the PCD preconditioner is the requirement to assemble an additional matrix, at an additional computational expense [128].

Another approach for the approximation of the Schur complement is using a least squares commutator (LSC) preconditioner [65], where the approximation is

$$\boldsymbol{M}_S = (B\boldsymbol{T}^{-1}B^T)(B\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}^{-1}B^T)^{-1}(B\boldsymbol{T}^{-1}B^T).$$

Here $\boldsymbol{T} = \operatorname{diag}(\boldsymbol{Q})$ is the diagonal of the velocity mass matrix. This approach is advantageous since it does not require the construction of additional operators such as F_p and $\boldsymbol{A}_p$. Implementing this approximation requires the inverse

$$\boldsymbol{M}_S^{-1} = (B\boldsymbol{T}^{-1}B^T)^{-1}(B\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}^{-1}B^T)(B\boldsymbol{T}^{-1}B^T)^{-1},$$

which does not require an approximation to the inverse of $\boldsymbol{A}$. When performing simple tests of two-dimensional problems with GMRES, the LSC generally outperforms (in terms of iterations to convergence) the PCD preconditioner [66], although at higher Reynolds numbers both approaches become more evenly matched.

To deal with inverting the convection-diffusion operator entry $\boldsymbol{A}^{-1}$ of the preconditioner, a cost-effective approach to approximate this could involve a fixed number of steps of an iterative method generating an appropriate inaccurate, but cheap, representation. For example, using a fixed number of multigrid cycles is a practical choice. It is observed that only a small increase in the iteration count is produced when replacing a direct solve with an algebraic multigrid (AMG) approach, yet it is significantly less expensive [66, Ch. 9].

Another approximation for $\boldsymbol{M}_A$ considers splitting $\boldsymbol{A} = \boldsymbol{D} - \boldsymbol{E}$, where $\boldsymbol{D}$ is an easily invertible matrix [51]. This approach is particularly effective when $\boldsymbol{A}$ is diagonally dominant. In the case where $\mathcal{A}$ is a symmetric saddle point system, with $\boldsymbol{A}$ being positive definite and $C = 0$, Fischer et al. [74] proposes a block preconditioner of the form,

$$M^{\pm} = \begin{bmatrix} \eta^{-1}\boldsymbol{A} & 0 \\ 0 & \pm M_S \end{bmatrix}.$$

Here η is a positive nonzero scaling parameter, and M_S is a symmetric positive definite approximation to the (negative) Schur complement, with the plus-minus giving the choice between a positive definite or indefinite matrix. This form of preconditioner is only relevant when $\boldsymbol{A}$ is easy to invert and therefore not applicable to discrete forms of the Navier-Stokes equation.

Both Elman [67] and Gauthier et al. [78] propose a preconditioner of a similar structure, with the latter deriving an inexact factorisation using a pressure matrix method (splitting velocity and pressure computations), whereas the former is designed with efficiency for Stokes and Oseen problems in mind.

Benzi and Golub [22] consider a different approach to preconditioning a generalised saddle point system, their approach is based upon alternating symmetric/skew-symmetric splitting scheme of the block system, in reference to the Oseen equations. The approach does not require the velocity block to be either nonsingular or symmetric, and therefore is applicable to a range of problems. They proposed that this stationary iteration is a competitive preconditioning approach for an asymmetric Krylov subspace solver, such as GMRES, rather than being used directly as an iterative method [10].

2.4.4 Iterative methods for FSI

As discussed previously one of the key problems when using monolithic approaches for fluid-structure interaction problems is to efficiently solve a large linearised system of algebraic equations at each time step [73, 146], the size of which can be compounded when considering three-dimensions [56].

Mori and Peskin [144] test a number of Krylov subspace iterative methods for a semi-implicit and fully implicit immersed boundary method. GMRES is found to be the most efficient solver for both schemes, where a conjugate gradient squared (CGS) method is found to fail to converge for the semi-implicit approach.

The work of Badia et al. [7, 8] focusses on designing a preconditioning approach for an ALE FSI scheme. They use an inexact factorisation by splitting the FSI problem into explicit and implicit steps, using a partitioned approach. The preconditioner is based upon inexact factorisation of the coefficient matrix, by using approximations of the Schur complements. By decoupling the velocity from the fluid-structure system, the pressure is still coupled to structural unknowns, ensuring that the scheme does not suffer from the added mass effect [8]. A comparison is conducted between an ILU, with threshold, and a split operator preconditioner using both GMRES and BiCGSTAB solvers for a monolithic formulation. Incomplete LU factorisation with threshold (ILUT) uses a numerical drop-tolerance instead of the sparsity pattern (standard ILU). The ILUT preconditioner is found to be more efficient in terms of the number of iterations and the solve time [7].

In a similar sense Heil [95], applied a block-triangular preconditioner to the monolithic Jacobian matrix, separating the coupling terms of fluid and structure. Thus, the preconditioners are applied through a number of substeps, solving four linear systems and up to three matrix vector products.

For applications of block preconditioning for FSI, an example uses a physics-based block preconditioning approach using AMG to approximate the inverse of diagonal blocks, where the blocks of the system are determined by physical fields; solid, fluid or ALE unknowns [136], as opposed to a more conventional block structure by velocity and pressure variables. This allows the approximate block inverses to be tailored to specific properties of the individual fields. However, this preconditioning approach is deficient in coupling between fields, where the error on the FSI interface is only reduced during outer iterations [80].

Gee et al. [80] consider two approaches. The first method considered a block preconditioned Newton-Krylov scheme, using different types of AMG tailored to the discretisation of each block. However, they note that the coupling between fields only occurs in the block Gauss-Seidel and Krylov method, with these approaches reducing only the interface errors. The second approach considers an AMG approximation for the entire coupled system, with prolongation and restriction operators tailored to each field. As a result, coupling between fields at the interface occurs not only on the fine grid block Gauss-Seidel but also at the coarse level of the approximation.

In a partitioned approach to solve an ALE FSI problems, Langer and Yang [125] use AMG extensively as an approximation to the inverse of discrete operators in preconditioned. In the approximation for the Schur complement for the fluid sub-problem, $\rho_f/\Delta t D_p^{-1} + \mu M_p^{-1} + \rho_f M_p^{-1} C_p D_p^{-1}$. Here D_p and C_p are stiffness matrices associated with the discretisation of the Laplacian and scalar convection operation in the pressure space, respectively. The discrete pressure mass matrix is replaced with $\operatorname{diag}(M_p)$ and AMG is used to approximate the action of the inverse of D_p. In both the fluid and solid subsystem, AMG cycles are used to obtain the inverse of the velocity block (coefficient matrix block acting on the velocities). They do make the explicit observation that AMG fails to solve hyperelastic problems, as a result of the off-diagonal contribution from coupling of the velocity components.

Muddle et al. [146] consider a block preconditioning approach which applies a “pseudo-solid” body representing the fluid domain and a fluid-loaded solid body. The fluid-loaded solid is modelled using Kirchhoff-Love thin-shell theory and assumed to be linearly elastic. For saddle-point systems, they make use of a specific kind of augmented preconditioner based on the least squares commutator (LSC) preconditioner as originally proposed by Elman et al. [65], which is applied in the case where the velocity block of the solution system is singular [86]. When preconditioning, they apply AMG to approximate the inverse of the principal diagonal and approximate Schur complement, with near-optimal computational cost. In their LSC preconditioner, for the approximation to the Schur complement they consider a pressure-Poisson like matrix which involves just the diagonal of the velocity mass matrix [146].

Motivated by FSI haemodynamics, Deparis et al. [56] use a block preconditioning technique with a SIMPLE preconditioner, after applying a static condensation of the fluid variables at the interface resulting in a reduced fluid matrix. A Gauss-Seidel factorisation of the FSI matrix based on a preconditioner applied to each physical component of the system (fluid, structure, and geometry). They only consider a Taylor-Hood finite element space in the discretisation, and they do not address the discontinuities at the interface in a specific manner.

Kong and Cai [121] examine the use of a multilevel, smoothed Schwarz preconditioner for a three-dimensional, unstructured mesh, monolithic FSI problem, applying incomplete LU factorisation as a subdomain solver. They use an equal-order, linear, stabilised finite element space for discretisation of the Navier-Stokes equation. The hierarchies for the multilevel Schwarz preconditioner are determined by a geometric multilevel method, which uses an isogeometric coarsened mesh to preserve geometry of the boundaries and the interface, instead of using an algebraic multigrid approach, which is typically applied to unstructured meshes. Overall, they find that their algorithm is scalable to 10,000 cores.

When preconditioning a nonlinear elastic solid, a fixed number of multigrid cycles is an attractive choice to approximate the velocity block [38, 204]. However, for higher-order elements, it can be expensive to assemble the required matrices. To avoid this, a p-multigrid preconditioning technique is applied [34], reducing the higher order element to linear brick elements Q_1, where AMG is then applied, in order to reduce memory requirements [138].

As with any monolithic, velocity-pressure coupled system, there are two main challenges to preconditioning: An efficient preconditioning to the velocity block; and, a good approximation to the Schur complement. Clearly, the adaptability of using AMG as a preconditioner for the velocity block is evident. However, when applying an “out-of-the-box” approach, it can become inefficient due to the coupling block of the off-diagonal velocity components, particularly when applied to hyperelastic solids. For FSI, the spectrally equivalent pressure-mass matrix is no longer guaranteed to be an appropriate approximation to the Schur complement.

2.5 FreeFEM software

All numerical experiments in this thesis were performed using the open source multiphysics PDE solver package, FreeFEM [93]. FreeFEM is high-level, typed, polymorphic, scripting language where the user outlines the variational form of the problem, and the software generates a discrete linear system using finite elements. It also has the ability to call a number of external packages for the mesh generation and solvers for the linear system of equations. Specifically, the packages that have been used in this project are outlined in Figure 2.2. It also offers flexibility when implementing a bespoke algorithm pipeline, using a C++ idiomatic framework to call routines when required.

FreeFEM has access to the sparse direct linear system solver, UMFPACK [48], which uses an unsymmetric multifrontal method. FreeFEM is also interfaced with PETSc [12], a set of libraries that provides access to a range of sequential and parallel, spare direct or iterative solvers. Packaged within PETSc are several additional libraries; these include the high performance AMG solver and preconditioning package, HYPRE (high performance preconditioners) [69]. One of the more powerful of these is an algebraic multigrid method, BoomerAMG [166]. In addition, Euclid [110] offers extensive ILU functionality, with the ability to perform an incomplete LU factorisation, with a range of thresholding options.

In two-dimensional FreeFEM, meshing is performed using an in-built triangular unstructured algorithm. For three-dimensional geometries, the TetGen [174] package is called within FreeFEM to construct unstructured tetrahedral meshes.

One of the key benefits of using FreeFEM is the ability to change the finite element approximation and solver from a number of built-in options within a few lines of code.

Unlike using a custom finite element code using C/C++ where the user has exact control over the allocation and handling of memory, FreeFEM presents a challenge when performing memory management. This can manifest itself as a large memory overhead compared to custom application codes.

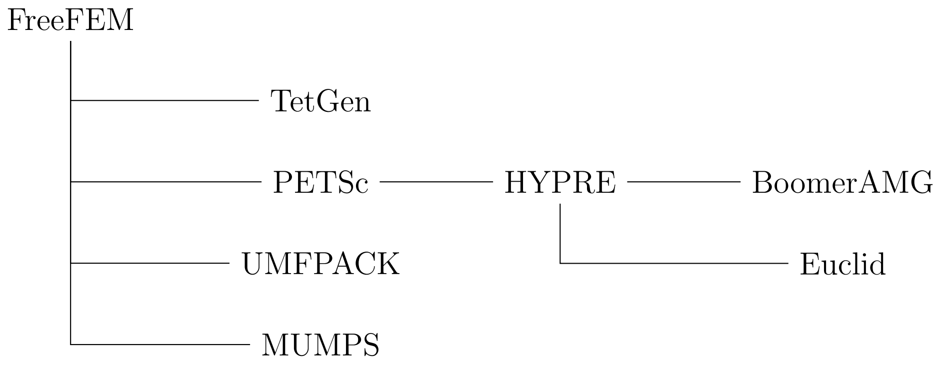


Figure 2.2: Hierarchy outlining the libraries within the FreeFEM package utilised in this project.

2.6 Concluding remarks

In general, the finite difference method is not appropriate for geometries that are not regular, a requirement which is of particular importance for deforming FSI. Although originally finite volume was designed with the intent of solving fluid problems and finite elements for structural mechanics, FEM's natural treatment of Neumann boundary conditions and its use on unstructured grids makes it particularly attractive to both [44]. A mesh-fitted approach lends itself well to the ALE approach, when explicitly tracking the interface between solid and fluid. Furthermore, using a single mesh simplifies the computational load (no overlapping meshes) and improves robustness, with no requirement to transfer information between meshes. However, this ALE conformal meshing approach is limited to relatively small deformations of the solid. Using a monolithic formulation proves a more stable approach for FSI, yet here lies the preconditioning challenge. With partitioned approaches, existing preconditioning methods, particularly for the fluid component, are well-researched.

Algebraic multigrid is a common tool when approximating the inverse of specific blocks within a block preconditioner. Although multigrid methods make effective solvers for linearised elliptic problems, in general they demonstrate poor scalability as a solver for multiphysics, but are much more effective when used to generate a preconditioner for Krylov subspace iterative solvers [97]. One of the key benefits of using AMG is the range of options which can control a varying discrete degrees of approximations. For example, by choosing the cycle types, number of multigrid levels and the maximum level of coarsening, the appropriate compromise between cost and accuracy of approximation can be found.

A key property of a good preconditioner for this FSI problem is an accurate spectral approximation of the Schur complement. Much existing work has considered this for the different discrete fluid systems. However, with the Schur complements dependency on the inverse of the velocity block we will need examine how the contribution from the solid in a monolithic formulation affects the performance.

This thesis will aim to use a monolithic ALE formulation to model some numerical test cases, based on the work by Wang et al. [201]. We construct the discrete system using FreeFEM to develop and test sparse direct solvers in two dimensions, and continue to use block preconditioned sparse iterative solvers in three dimensions. Overall, the monolithic formulation should provide a stable and accurate solution, with a block preconditioner providing a more efficient solution method.

Chapter 3

An Arbitrary Lagrangian Eulerian method for FSI

This section outlines the kinematics of the Arbitrary Lagrangian Eulerian (ALE) description for the specific method used within this thesis. A fully Lagrangian approach moves the computational mesh to follow the velocity of the fluid [101]. This offers a computational advantage, since each finite element cell of a Lagrangian mesh will always contain the same material particles. This is suitable for low Reynolds problems and (near) irrotational flow. Beyond this regime, without remeshing, the quality of the deforming mesh will deteriorate rapidly over time. In contrast, the Eulerian approach for finite elements is particularly popular for purely fluid problems. Here, the computational mesh is fixed in time, with the continuum moving and deforming with respect to the mesh. Eulerian formulation can present a numerical challenge for convection-dominated problems due to the non-symmetric nature of convection operators [58].

The ALE approach provides a compromise between Lagrangian and Eulerian methods. The interface between the solid and fluid is explicitly tracked as it deforms, the surrounding mesh is able to deform in a manner that aims to maintain mesh quality (as opposed to following the materials points, as in a Lagrangian frame). The freedom to move the surrounding mesh allows greater distortion of the continuum (fluid and solid) that would be permitted in a purely Lagrangian scheme. The ALE scheme therefore does require an extra mechanism for controlling the deformation of the mesh, which is separate from the control equations for continua.

Even with a robust scheme for maintaining mesh quality, the ALE method is limited to small deflections before the mesh becomes too deformed and loses accuracy. At which point, a total remesh is required, which involves interpolation of the solution onto the new mesh. Another drawback, which is relevant to the Lagrangian mesh approaches, is that the coefficient matrix must be recalculated at each time step the mesh is deformed. The geometric connectivity between degrees of freedom will change during the mesh deformation, thus the shape functions must be recalculated. A fixed mesh within a Eulerian sense does not have this problem.

The ALE framework described in this section is based on work by Wang et al. [201], where the extension to a three-dimensional neo-Hookean solid is taken from Chiang et al. [43].

3.1 ALE kinematics

We first define the time-dependent fluid and solid domains, $\Omega_t^f \in \mathbb{R}^d$ and $\Omega_t^s \in \mathbb{R}^d$, respectively. Using the superscript notation of f and s for the fluid and solid variables throughout, and the superscript $d = 2, 3$ is the dimensionality of the problem. The interface between the fluid and solid regions is denoted by $\Gamma_t = \Omega_t^f \cap \Omega_t^s$, as shown in Figure 3.1. On the contrary, the total domain is defined as $\Omega_t = \Omega_t^f \cup \Omega_t^s$, which has an outer boundary of $\partial\Omega_t$.

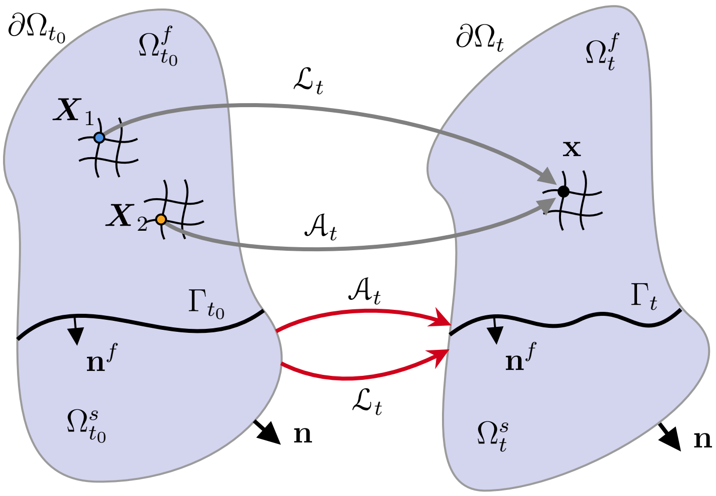


Figure 3.1: ALE mapping from domain Ω_{t_0} to Ω_t, with the Lagrangian mapping $\mathcal{L}_t$ and ALE mapping $\mathcal{A}_t$.

As discussed previously, an Eulerian frame is useful for describing a fluid in a fixed frame, in this formulation the Lagrangian frame lends itself to describing the solid as it moves with this frame. Typically, in ALE formulations, the solid equation is solved in the reference configuration on a fixed mesh, where the ALE transformation is only applied to the fluid. Therefore, the ALE frame can be useful within the FSI context, when both the solid and fluid regions interact through the interface. We consider a frame that can move arbitrarily from the material domain Ω_{t_0}, the undeformed reference state, to the spatial domain Ω_t, which is the current configuration. A family of mappings are defined as,

$$\mathcal{A}_t : \Omega_{t_0} \to \Omega_t. \tag{3.1}$$

It is assumed that this family of maps, $\mathcal{A}_t \in C^0(\Omega_{t_0})^d$ are continuous, one to one mapping. It also possesses a continuous inverse mapping, $\mathcal{A}_t^{-1} \in C^0(\Omega_t)^d$. Thus, a point $\boldsymbol{X} \in \Omega_{t_0}$ has a unique image at $\boldsymbol{x} \in \Omega_t$,

$$\boldsymbol{x} = \mathcal{A}(\boldsymbol{X}, t) = \mathcal{A}_t(\boldsymbol{X}). \tag{3.2}$$

Similarly for the inverse mapping,

$$\boldsymbol{X} = \mathcal{A}^{-1}(\boldsymbol{x}, t) = \mathcal{A}_t^{-1}(\boldsymbol{x}). \tag{3.3}$$

It is assumed that the mapping, $\mathcal{A}(\boldsymbol{X}, t)$ is differentiable in time giving a velocity for the ALE frame,

$$\boldsymbol{w}(\boldsymbol{X}, t) = \frac{\partial \mathcal{A}}{\partial t}(\boldsymbol{X}, t). \tag{3.4}$$

It is this ALE mapping which is used to map the deformation of the domain as the solid deforms, therefore this velocity will often be referred to as the mesh velocity (indicating that the discretised system is solved on a moving finite element mesh). The material coordinate maps to the Eulerian coordinate $\boldsymbol{x} \in \Omega_t$, via a Lagrangian mapping,

$$\mathcal{L}_t : \boldsymbol{X} \mapsto \boldsymbol{x} \tag{3.5}$$

$$\boldsymbol{x} = \mathcal{L}(\boldsymbol{X}, t) = \mathcal{L}_t(\boldsymbol{X}), \tag{3.6}$$

which is the trajectory of the material particle. The velocity of the Lagrangian mapping at $\boldsymbol{X}$ is given as,

$$\boldsymbol{u}(\boldsymbol{X}, t) = \frac{\partial \mathcal{L}}{\partial t}, \tag{3.7}$$

which is the material velocity. The Lagrangian and ALE mappings are not generally the same. Both mappings are constructed so that at the fluid-solid interface they coincide, $\mathcal{A}_t(\Gamma_t) = \mathcal{L}_t(\Gamma_t)$.

3.2 ALE formulation for FSI

We begin with the momentum equation which takes the same form for both, in the current configuration, Ω_t,

$$\rho \frac{d\boldsymbol{u}}{dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{g}, \tag{3.8}$$

where ρ, $\boldsymbol{u}$, $\boldsymbol{\sigma}$ and $\boldsymbol{g}$ are the density, material velocity, Cauchy stress tensor and gravity respectively. Here, $d(\cdot)/dt$ is the total derivative (or material derivative) along the trajectory of the particle at $\boldsymbol{x}$.

The mass continuity equation for the incompressible material in the deformed state is

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in} \quad \Omega_t. \tag{3.9}$$

Now we consider the constitutive relationship for both an incompressible fluid and incompressible solid, expanding the Cauchy stress tensor,

$$\boldsymbol{\sigma} = \boldsymbol{\tau} - p\boldsymbol{I} \tag{3.10}$$

where $\boldsymbol{\tau}$ is the deviatoric component of the stress tensor. An incompressible solid assumption is considered, which will be addressed later.

For a incompressible Newtonian viscous fluid, $\boldsymbol{\sigma}|_{\Omega_t^f} = \boldsymbol{\tau}^f + p^f\boldsymbol{I}$, in the fluid domain,

$$\boldsymbol{\tau}^f = \mu^f \mathrm{D}\boldsymbol{u} = \mu^f(\nabla\boldsymbol{u} + \nabla^T\boldsymbol{u}) \quad \text{in} \quad \Omega_t^f. \tag{3.11}$$

The expanded notation $\mathrm{D}(\cdot) = \nabla(\cdot) + \nabla^T(\cdot)$ and μ^f is the dynamic viscosity of the fluid.

Similarly, for a hyperelastic solid, $\boldsymbol{\sigma}|_{\Omega_t^s} = \boldsymbol{\tau}^s + p^s\boldsymbol{I}$, in the solid domain,

$$\boldsymbol{\tau}^s = J_{\mathcal{L}_t}^{-1}\boldsymbol{P}\boldsymbol{F}^T \quad \text{in} \quad \Omega_t^s, \tag{3.12}$$

where the first Piola-Kirchhoff stress tensor is $\boldsymbol{P} = \partial\Psi(\boldsymbol{F})/\partial\boldsymbol{F}$. The strain energy of a hyperelastic solid, $\Psi(\boldsymbol{F})$, is a function of the deformation tensor of the solid, which is defined as,

$$\boldsymbol{F} = \frac{\partial\boldsymbol{x}}{\partial\boldsymbol{X}} = \frac{\partial\mathcal{L}(\boldsymbol{X},t)}{\partial\boldsymbol{X}}, \tag{3.13}$$

which can also be written as $\boldsymbol{F} = \nabla_X^T\boldsymbol{x}$. The Jacobian is defined as the determinant of the deformation tensor, $J_{\mathcal{L}_t} = \det(\boldsymbol{F})$.

Finally we define the continuity of the velocity and normal stress across the interface for the FSI system,

$$\boldsymbol{u}^f = \boldsymbol{u}^s \quad \text{on} \quad \Gamma_t, \tag{3.14a}$$

$$\boldsymbol{\sigma}^f\boldsymbol{n}^f = -\boldsymbol{\sigma}^s\boldsymbol{n}^s \quad \text{on} \quad \Gamma_t. \tag{3.14b}$$

Dirichlet and Neumann boundary conditions are imposed on the fluid, generally

$$\boldsymbol{u}^f = \overline{\boldsymbol{u}} \quad \text{on} \quad \Gamma_D, \tag{3.15a}$$

$$\boldsymbol{\sigma}^f\boldsymbol{n} = \overline{\boldsymbol{h}} \quad \text{on} \quad \Gamma_N. \tag{3.15b}$$

3.2.1 Transformation to the ALE frame

The material derivative computed along the trajectory of a material particle is expanded using the Lagrangian mapping

$$\frac{d\boldsymbol{u}(\boldsymbol{x},t)}{dt} = \frac{d\boldsymbol{u}(\mathcal{L}_t(\boldsymbol{X}),t)}{dt} = \left.\frac{\partial \boldsymbol{u}}{\partial t}\right|_{\boldsymbol{x}=\mathcal{L}_t(\boldsymbol{X},t)} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}. \tag{3.16}$$

Where the above partial derivative is replaced with the total derivative of the ALE mapping

$$\frac{d\boldsymbol{u}(\mathcal{A}_t(\boldsymbol{X},t)}{dt} = \left.\frac{\partial \boldsymbol{u}}{\partial t}\right|_{\boldsymbol{x}=\mathcal{A}_t(\boldsymbol{X},t)} + (\boldsymbol{w}\cdot\nabla)\boldsymbol{u}. \tag{3.17}$$

Substituting into equation (3.8) formulates the ALE momentum equation,

$$\rho\frac{d\boldsymbol{u}(\mathcal{A}_t(\boldsymbol{X},t))}{dt} + \rho((\boldsymbol{u}-\boldsymbol{w})\cdot\nabla)\boldsymbol{u} = \nabla\cdot\boldsymbol{\sigma} + \rho\boldsymbol{g}. \tag{3.18}$$

We note that the mass continuity equation (3.9) is the same in the ALE frame. We will return to this in Section 3.4 when considering the choice of the mesh velocity.

3.3 Hyperelastic solid model

For larger deforming solids, the linear stress-strain relationship of linear elastic models will not accurately represent the physics, whereas a hyperelastic solid model is able to more accurately capture the complex physics with large deflections, under the constraint that the ALE is able to maintain sufficient mesh quality.

Hyperelastic solids are constitutive models of elastic solids where the stress-strain relationship is derived from a strain energy function; they aim to capture the nonlinear stress-strain curve for a soft rubber-like material. Early invariant-based models include neo-Hookean [190] and Mooney-Rivlin [143] solids. These are simple one-variable models and are considered a baseline for rubber elastic models, however, the stress response plateau at large strains limiting them to small, to moderate, deformations. Further models formulated with invariance include the Gent [81] or Yeoh [208] models. Principle stress models could include Valanis-Landel [195] and Ogden [150], and another family of models is based on the statistical mechanics or micromechanics of the material, such as the Arruda-Boyce model [4, 47].

3.3.1 Neo-Hookean formulation

A neo-Hookean model offers a fairly simple nonlinear solid model, with only a single material constant for an incompressible solid. The strain energy density function, also known as the Helmholtz free energy per unit reference volume function, for a general compressible neo-Hookean model [29, Ch. 6.4.3] [98, 201], is

$$\boldsymbol{\Psi}(\boldsymbol{F}) = c_1(I_1 - d) - 2c_1 \ln J_{\mathcal{L}_t} + k \ln^2 J_{\mathcal{L}_t}. \tag{3.19}$$

The two material constants are $c_1 = \mu^s/2$ and $k = \lambda^s/2$, λ^s and μ^s are the first and second (often referred to as the shear modulus) Lamé parameters. Again, d is the dimension of the problem.

The principle invariants in terms of the left Cauchy-Green deformation tensor, $\boldsymbol{B}$, are

$$I_1 := \mathrm{tr}(\boldsymbol{B}) = \mathrm{tr}(\boldsymbol{F}\boldsymbol{F}^T) \tag{3.20}$$

$$I_2 := \frac{1}{2}\left(\mathrm{tr}(\boldsymbol{B})^2 - \mathrm{tr}(\boldsymbol{B}^2)\right) = \frac{1}{2}\left(\mathrm{tr}(\boldsymbol{F}\boldsymbol{F}^T)^2 - \mathrm{tr}(\boldsymbol{F}\boldsymbol{F}^{T^2})\right). \tag{3.21}$$

Compared to a compressible Mooney-Rivlin solid (a more general form of a neo-Hookean solid [151, Ch. 4.3])

$$\boldsymbol{\Psi}(\boldsymbol{F}) = c_{10}(I_1 - d) + c_{01}(I_2 - d) + k \ln^2 J_{\mathcal{L}_t}, \tag{3.22}$$

where $(c_{10} + c_{01}) = \mu^s/2$. Both these models are special cases of the general polynomial rubber elastic solids, which just consists of a summation of the product of the two invariants, I_1 and I_2, to any given power [30, Ch. 3.5].

Differentiating equation (3.19) with respect to the deformation tensor, requires two identities. Firstly, making use of a special case of Jacobi's formula [133], the partial derivative is

$$\frac{\partial J_{\mathcal{L}_t}}{\partial \boldsymbol{F}} = J_{\mathcal{L}_t}\, \mathrm{adj}^T(\boldsymbol{F}) = J_{\mathcal{L}_t} \boldsymbol{F}^{-T}. \tag{3.23}$$

And secondly, the differential of the trace of the gradient tensor and its transpose product, $\partial(\mathrm{tr}(\boldsymbol{F}\boldsymbol{F}^T))/\partial\boldsymbol{F} = 2\boldsymbol{F}$. Applying these to the strain energy function for the neo-Hookean solid in (3.19)

$$\frac{\partial\Psi(\boldsymbol{F})}{\partial\boldsymbol{F}} = \mu^s(\boldsymbol{F} - \boldsymbol{F}^{-T}) + \lambda^s \ln{(J_{\mathfrak{L}_t})}\boldsymbol{F}^{-T}. \tag{3.24}$$

Multiplying the above through by $\boldsymbol{F}^T$

$$\frac{\partial\Psi(\boldsymbol{F})}{\partial\boldsymbol{F}}\boldsymbol{F}^T = \mu^s(\boldsymbol{F}\boldsymbol{F}^T - \boldsymbol{I}) + \lambda^s \ln{(J_{\mathfrak{L}_t})}\boldsymbol{I}. \tag{3.25}$$

And then substituting this into equation (3.12),

$$\boldsymbol{\tau}^s = \mu^s J_{\mathfrak{L}_t}^{-1}(\boldsymbol{F}\boldsymbol{F}^T - \boldsymbol{I}) + \lambda^s J_{\mathfrak{L}_t}^{-1} \ln{(J_{\mathfrak{L}_t})}\boldsymbol{I}. \tag{3.26}$$

For an incompressible solid, $J_{\mathfrak{L}_t} = 1$ therefore in the last term $\ln(J_{\mathfrak{L}_t}) = 0$, thus,

$$\boldsymbol{\tau}^s = \mu^s J_{\mathfrak{L}_t}^{-1}(\boldsymbol{F}\boldsymbol{F}^T - \boldsymbol{I}). \tag{3.27}$$

The last term $\mu^s J_{\mathfrak{L}_t}^{-1}\boldsymbol{I}$ is often replaced by a pressure term $-p'\boldsymbol{I}$ that is absorbed into the solid pressure. The divergence-free velocity imposed by $\nabla \cdot \boldsymbol{u}^s = 0$ is used to constrain this solid pressure [5]. For completeness, for an incompressible neo-Hookean solid stress is given as,

$$\boldsymbol{\sigma}^s = \boldsymbol{\tau}^s - p^s\boldsymbol{I} = \mu^s J_{\mathfrak{L}_t}^{-1}(\boldsymbol{F}\boldsymbol{F}^T - \boldsymbol{I}) - p^s\boldsymbol{I}. \tag{3.28}$$

This can be taken even further to include two terms into the pressure, $(p' + p^s)\boldsymbol{I} = \mu^s J_{\mathfrak{L}_t}^{-1}\boldsymbol{I} + p^s\boldsymbol{I}$ [201]. Since the solid is not stress-free, this will generate a pressure jump at the fluid-solid interface.

3.4 Mesh movement

Within the ALE framework, the computational mesh must move in a manner that maintains element quality so that the solution to the FSI system is accurate. It is important to consider a computationally efficient approach, since the accuracy in the solution to the mesh movement calculation is not essential in itself, however, will inherently affect the solid deformation. Further, with large deformations of the solid, one would aim to minimise the use of remeshing, which tends to be particularly computationally expensive.

In terms of solving a partial differential equation to control the movement of the mesh, a number of different approaches are available. Refer to Section 2.3.2 for further discussion of different mesh movement approaches. A linear elastic model offers a sufficiently robust and responsive model (to the fluid-sold interface displacement) that is simple to solve (owing to its elliptic properties).

As the computational mesh tracks the interface between the solid and fluid, the movement of the mesh must conform to the movement of this interface. The surrounding mesh will neither move in a purely Lagrangian or Eulerian description, and hence the term arbitrary Lagrangian Eulerian. A fairly intuitive approach is to consider the mesh as an elastic material which is able to deform under force from the fluid-solid interface and return to its original shape.

In a similar approach to Johnson and Tezduyar [113] a mesh moving scheme involving the motion of the nodes is controlled by the equations of linear elasticity. A separate system is considered in the form of a static linear elastic material of the form,

$$\nabla \cdot (\mu^m \mathrm{D}\boldsymbol{w} + \lambda^m (\nabla \cdot \boldsymbol{w})\boldsymbol{I}) = 0. \tag{3.29}$$

Where λ^m and μ^m are artificial first and second Lamé parameters respectively, determining the response of the mesh [31, Ch. 11.1], which are typically different from the material properties of the solid. Usually, linear elastic equations are solved for displacement; however, in this case, the unknown is $\boldsymbol{w} \in V^h(\Omega_t)$, the ALE velocity introduced previously. Since it is a linear constitutive relationship $\boldsymbol{w} \approx \Delta \boldsymbol{d}/\Delta t$, the relationship with velocity should only scale with the reciprocal of the time step. We note that solving a linear elastic system for velocity, to control the movement of the mesh, is first applied by Wang et al. [201].

The boundary conditions are given by

$$\boldsymbol{w} \cdot \boldsymbol{n} = 0, \ \boldsymbol{n} \cdot \mathrm{D}\boldsymbol{w} \cdot \boldsymbol{t} = 0, \quad \text{on} \quad \partial\Omega_t \tag{3.30}$$

where $\boldsymbol{n}$ and $\boldsymbol{t}$ are the normal and tangential directions, respectively, of the boundary $\partial\Omega$, permitting the mesh nodes to move tangentially to the boundary surface. The enforcement of $\boldsymbol{n}\cdot \mathrm{D}\boldsymbol{w}\cdot \boldsymbol{t} = 0$ ensures a unique solution to the elliptic problem in (3.29) [120, 201]. The interface condition

$$\boldsymbol{w} = \boldsymbol{u}, \quad \text{on} \quad \Gamma_t, \tag{3.31}$$

permits the mesh nodes to move with the true fluid velocity.

3.5 Weak ALE formulation

The next section is based upon the derivation of the ALE formulation from Wang et al. [201]. From the mapping described in Equation (3.2), any function $f \in H_0^1(\Omega_t)$ in the current configuration, one-to-one correspondence to the function $\hat{f} \in H_0^1(\Omega_{t_0})$ in the reference configuration, by $\hat{f} = f \circ \mathcal{A}_t$. Let $L^2(\Omega) = \{f : \int_\Omega |f|^2 < \infty\}$ be a square-integrable function, and $H^1(\Omega) = \{f : \int_\Omega |f|^2 < \infty, \int_\Omega ||\nabla f||^2 < \infty\}$ also require the square-integrable gradient of the function, both in the domain Ω. Denoting the test space $H_0^1(\Omega)$ as a subspace of $H^1(\Omega)$, whose functions are zero on the Dirichlet boundary Ω.

To derive the weak formulation of the ALE system equation, (3.18) is multiplied by a test function $\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{v} \circ \mathcal{A}_t(\boldsymbol{X}) = \boldsymbol{V}(\boldsymbol{X})$ and integrated in the respective domains, given the test space $\boldsymbol{v} \in H_0^1(\Omega_t)^d$. By integrating the stress term by parts,

$$\begin{aligned}\rho^f \int_{\Omega_t^f} \frac{d\boldsymbol{u}(\mathcal{A}_t(\boldsymbol{X},t))}{dt} \cdot \boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \rho^f \int_{\Omega_t^f} ((\boldsymbol{u}-\boldsymbol{w})\cdot\nabla)\boldsymbol{u}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \int_{\Omega_t^f} \boldsymbol{\sigma}^f : \mathrm{D}\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} \\ = \int_{\partial\Omega_t^f} \boldsymbol{\sigma}^f \boldsymbol{n}^f \cdot \boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \rho^f \int_{\Omega_t^f} \boldsymbol{g}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} \quad \text{in} \quad \Omega_t^f.\end{aligned} \tag{3.32}$$

In the above equation we consider the standard double dot notation for general tensors, $\boldsymbol{A} : \boldsymbol{B} = \sum_{i,j=1}^d A_{ij} B_{ij}$. In the solid domain,

$$\begin{aligned}\rho^s \int_{\Omega_t^s} \frac{d\boldsymbol{u}(\mathcal{A}_t(\boldsymbol{X},t))}{dt} \cdot \boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \rho^s \int_{\Omega_t^s} ((\boldsymbol{u}-\boldsymbol{w})\cdot\nabla)\boldsymbol{u}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \int_{\Omega_t^s} \boldsymbol{\sigma}^s : \mathrm{D}\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} \\ = \int_{\partial\Omega_t^s} \boldsymbol{\sigma}^s(-\boldsymbol{n}^f) \cdot \boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \rho^s \int_{\Omega_t^s} \boldsymbol{g}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} \quad \text{in} \quad \Omega_t^s.\end{aligned} \tag{3.33}$$

The following integral is transformed from the current configuration to the reference configuration by,

$$\int_{\Omega_t^s} \boldsymbol{\tau}^s : \nabla \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} = \int_{\Omega_t^s} J_{\mathcal{L}_t}^{-1} \frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{F}} \boldsymbol{F}^T : \nabla \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} = \int_{\Omega_{t_0}^s} \frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{F}} : \nabla_{\boldsymbol{X}} \boldsymbol{v} \, \mathrm{d}\boldsymbol{x}, \tag{3.34}$$

as derived in Hecht and Pironneau [94]. Using the interface conditions and the previous integral transformation applied to the solid stress in equation (3.12), equations (3.32) and (3.33) are summed,

$$\begin{aligned} \rho \int_{\Omega_t} \frac{d\boldsymbol{u}(\mathcal{A}_t(\boldsymbol{X},t))}{dt} \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} + \rho \int_{\Omega_t} ((\boldsymbol{u}-\boldsymbol{w}) \cdot \nabla)\boldsymbol{u} \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} + \frac{\mu^f}{2} \int_{\Omega_t^f} \mathrm{D}\boldsymbol{u} : \mathrm{D}\boldsymbol{v} \, \mathrm{d}\boldsymbol{x} \\ + \int_{\Omega_t} p \nabla \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} + \int_{\Omega_{t_0}^s} \frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{F}} : \nabla_{\boldsymbol{X}} \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} = \rho \int_{\Omega_t} \boldsymbol{g} \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} \qquad \text{in} \quad \Omega_t. \end{aligned} \tag{3.35}$$

The integrals at the interface Γ_t are cancelled by the condition (3.14b), since the system considers the entire net force of the total domain.

The weak form of the mass continuity equation, (3.9), is obtained by multiplying by the test function $q \in L^2(\Omega_t)$

$$-\int_{\Omega_t} q \nabla \cdot \boldsymbol{u} \, \mathrm{d}\boldsymbol{x} = 0 \qquad \text{in} \quad \Omega_t. \tag{3.36}$$

The mesh movement equation (3.29) is treated as a separate system and a different test function $\boldsymbol{z} \in H_0^1(\Omega_t)$ defined to derive the weak form. Integrating both terms by parts,

$$\frac{\mu^m}{2} \int_{\Omega_t} \mathrm{D}\boldsymbol{w} : \mathrm{D}\boldsymbol{z} \, \mathrm{d}\boldsymbol{x} + \lambda^m \int_{\Omega_t} (\nabla \cdot \boldsymbol{w})(\nabla \cdot \boldsymbol{z}) \, \mathrm{d}\boldsymbol{x} = 0. \tag{3.37}$$

The complete weak form of the FSI-ALE problem is outlined in Problem 1, and reads as follows.

Problem 1: Given Ω_{t_0}, Γ_{t_0}, $\boldsymbol{u}(\boldsymbol{X}, t_0)$ and the ALE mapping $\mathcal{A}_t$, find $\boldsymbol{u}(\boldsymbol{x}, t) \in H_0^1(\Omega_t)^d$ and $p(\boldsymbol{x}, t) \in L^2(\Omega_t)$, given the test functions $\forall \boldsymbol{v}(\boldsymbol{x}) \in H_0^1(\Omega_t)^d$ and $q \in L^2(\Omega_t)$, for the following equations

$$\begin{aligned} \rho \int_{\Omega_t} \frac{d\boldsymbol{u}(\mathcal{A}_t(\boldsymbol{X},t))}{dt} \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} + \rho \int_{\Omega_t} ((\boldsymbol{u}-\boldsymbol{w}) \cdot \nabla)\boldsymbol{u} \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} + \frac{\mu^f}{2} \int_{\Omega_t^f} \mathrm{D}\boldsymbol{u} : \mathrm{D}\boldsymbol{v} \, \mathrm{d}\boldsymbol{x} \\ + \int_{\Omega_t} p \nabla \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} + \int_{\Omega_{t_0}^s} \frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{F}} : \nabla_{\boldsymbol{X}} \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} = \rho \int_{\Omega_t} \boldsymbol{g} \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} \end{aligned} \tag{3.38}$$

and

$$-\int_{\Omega_t} q\nabla \cdot \boldsymbol{u} \, \mathrm{d}\boldsymbol{x} = 0. \tag{3.39}$$

Here, the ALE and Lagrangian mappings correspond at the boundary and interface, $\mathcal{A}_t(\partial\Omega_{t_0}) = \mathcal{L}_t(\partial\Omega_{t_0})$ and $\mathcal{A}_t(\Gamma_{t_0}) = \mathcal{L}_t(\Gamma_{t_0})$.

The discretisation of this problem, in time and space, is provided in the following sections.

3.6 Finite element discretisation

The weak formulations are discretised in time and space to form a system of equations to be solved at each time step, using a uniform time step of Δt.

3.6.1 Discretisation in time

To discretise in time an implicit Euler scheme is applied, which is first-order accurate.

From the definition of the deformation gradient tensor, $\boldsymbol{F}$, in (3.13), over a timestep Δt,

$$\frac{\boldsymbol{F}_{n+1} - \boldsymbol{F}_n}{\Delta t} = \frac{\boldsymbol{F}_{n+1} \circ \mathcal{L}_{t_{n+1}}(\boldsymbol{X}) - \boldsymbol{F}_n \circ \mathcal{L}_{t_n}(\boldsymbol{X})}{\Delta t} \approx \nabla_{\boldsymbol{X}} \boldsymbol{u}_{n+1}, \tag{3.40}$$

we rearranging these the deformation tensor is updated by

$$\boldsymbol{F}_{n+1} = \boldsymbol{F}_n + \Delta t \nabla_{\boldsymbol{X}} \boldsymbol{u}_{n+1}. \tag{3.41}$$

And from the ALE velocity, $\boldsymbol{w}$, in (3.4),

$$\frac{\boldsymbol{x}_{n+1} - \boldsymbol{x}_n}{\Delta t} = \frac{\mathcal{A}_{t_{n+1}}(\boldsymbol{X}) - \mathcal{A}_{t_n}(\boldsymbol{X})}{\Delta t} \approx \boldsymbol{w}_{n+1}. \tag{3.42}$$

the coordinates of the mesh are updated by

$$\Omega_{t_{n+1}} = \mathcal{A}_{t_{n+1}}(\Omega_{t_0}) = \{\boldsymbol{x} : \boldsymbol{x} = \boldsymbol{x}_n + \Delta t \boldsymbol{w}_{n+1}, \boldsymbol{x}_n \in \mathcal{A}_{t_n}(\Omega_{t_0})\}. \tag{3.43}$$

Given $\boldsymbol{u}_n = \boldsymbol{u}(\mathcal{A}_{t_n}(\boldsymbol{X}), t_n)$, finding $\boldsymbol{u}_{n+1} = \boldsymbol{u}(\mathcal{A}_{t_{n+1}}(\boldsymbol{X}), t_{n+1}) \in H_0^1(\Omega_{t_{n+1}})^d$ and $p(\mathcal{A}_{t_{n+1}}(\boldsymbol{X}), t_{n+1}) \in L^2(\Omega_{t_{n+1}})$, the time discretised momentum equation becomes,

$$\frac{\rho}{\Delta t}\int_{\Omega_{t_{n+1}}} \boldsymbol{u}_{n+1}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} - \frac{\rho}{\Delta t}\int_{\Omega_{t_{n+1}}} \boldsymbol{u}_{n}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \rho\int_{\Omega_{t_{n+1}}} (\boldsymbol{u}_{n+1}\cdot\nabla)\boldsymbol{u}_{n+1}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x}$$
$$- \rho\int_{\Omega_{t_n}} (\boldsymbol{w}_{n}\cdot\nabla)\boldsymbol{u}_{n+1}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \frac{\mu^f}{2}\int_{\Omega^f_{t_{n+1}}} \mathrm{D}\boldsymbol{u}_{n+1} : \mathrm{D}\boldsymbol{v}\,\mathrm{d}\boldsymbol{x}$$
$$+ \int_{\Omega_{t_{n+1}}} p_{n+1}\nabla\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \int_{\Omega^s_{t_0}} \frac{\partial\boldsymbol{\Psi}}{\partial\boldsymbol{F}}(\boldsymbol{F}_{n+1}) : \nabla_{\boldsymbol{X}}\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} = \rho\int_{\Omega_{t_{n+1}}} \boldsymbol{g}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x}. \quad (3.44)$$

and mass continuity equation,

$$-\int_{\Omega_{t_{n+1}}} q\nabla\cdot\boldsymbol{u}_{n+1}\,\mathrm{d}\boldsymbol{x} = 0. \quad (3.45)$$

It should be noted that the ALE velocity $\boldsymbol{w}_n$ from the previous time step will be found from the linear elastic equation (3.37) in Ω_{t_n}.

3.6.2 Discretisation in space

Firstly we define a stable finite element space (for example Taylor-Hood elements) to find the velocity and pressure solution in Ω_{t_0}, where the test functions belong to

$$\boldsymbol{v} \in V^h(\Omega_{t_0}) \subset H_0^1(\Omega_{t_0})^d, \quad (3.46)$$
$$q \in Q^h(\Omega_{t_0}) \subset L^2(\Omega_{t_0}). \quad (3.47)$$

As such in Ω_t,

$$V^h(\Omega_t) = \{\phi^h : \phi^h = \Phi\circ\mathcal{A}_t^{-1}, \Phi \in V^h(\Omega_{t_0}\} \quad (3.48)$$
$$Q^h(\Omega_t) = \{\phi^h : \psi^h = \Psi\circ\mathcal{A}_t^{-1}, \Psi \in Q^h(\Omega_{t_0}\}, \quad (3.49)$$

where the capital Greek letter notates the basis function in the ALE frame.

The discrete approximate solution to the velocity can be expressed in terms of the finite element basis functions for velocity, ϕ,

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{i=1}^{N^u} \boldsymbol{u}(\boldsymbol{x}_i)\phi_i(\boldsymbol{x}) + \sum_{i=N^u+1}^{N^u+N^u_D} \overline{\boldsymbol{u}}(\boldsymbol{x}_i)\phi_i(\boldsymbol{x}). \quad (3.50)$$

The number of unknown velocities to be found is denoted by N^u, and the number of discrete Dirichlet velocities on the boundary Γ_D is given by N^u_D.

The discrete approximate solution to the pressure is then expressed in terms of separate basis functions, ψ,

$$p^h(\boldsymbol{x}) = \sum_{i=1}^{N^p} p(\boldsymbol{x}_i)\psi_i(\boldsymbol{x}). \tag{3.51}$$

In this case, Dirichlet boundaries are not prescribed for the pressure solution, the total number of unknowns for pressure is N^p. For pressure spaces which involve a piecewise constant discontinuous basis, the unknowns are not located at the vertices of the element and will not be located on the boundary of the domain.

The space discretisation of equation (3.44) and (3.45) equations to find $(\boldsymbol{u}^h_{n+1}, p^h_{n+1}) \in V^h(\Omega_{t_{n+1}}) \times Q^h(\Omega_{t_{n+1}})$ given that $\boldsymbol{v} \in V^h(\Omega_{t_{n+1}})$ and $q \in Q^h(\Omega_{t_{n+1}})$, leads to

$$\begin{aligned} \frac{\rho}{\Delta t}\int_{\Omega_{t_{n+1}}} \boldsymbol{u}^h_{n+1}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} - \frac{\rho}{\Delta t}\int_{\Omega_{t_{n+1}}} \boldsymbol{u}^h_{n}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \rho\int_{\Omega_{t_{n+1}}} (\boldsymbol{u}^h_{n+1}\cdot\nabla)\boldsymbol{u}^h_{n+1}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} \\ - \rho\int_{\Omega_{t_{n+1}}} (\boldsymbol{w}^h_{n+1}\cdot\nabla)\boldsymbol{u}^h_{n+1}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \frac{\mu^f}{2}\int_{\Omega^f_{t_{n+1}}} \mathrm{D}\boldsymbol{u}^h_{n+1} : \mathrm{D}\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} \\ + \int_{\Omega_{t_{n+1}}} p^h_{n+1}\nabla\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \int_{\Omega^s_{t_0}} \frac{\partial\boldsymbol{\Psi}}{\partial\boldsymbol{F}}(\boldsymbol{F}_{n+1}) : \nabla_{\boldsymbol{X}}\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} = \rho\int_{\Omega_{t_{n+1}}} \boldsymbol{g}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x}, \end{aligned} \tag{3.52}$$

and,

$$-\int_{\Omega_{t_{n+1}}} q\nabla\cdot\boldsymbol{u}^h_{n+1}\,\mathrm{d}\boldsymbol{x} = 0. \tag{3.53}$$

The pressure p throughout the system is not uniquely determined, and can “float” by an undetermined constant and still be able to satisfy the governing equations. To avoid this an additional pressure constraint is applied [23, 25],

$$\int_{\Omega_{t_{n+1}}} p^h\,\mathrm{d}\boldsymbol{x} = 0. \tag{3.54}$$

Since the pressure in both the solid and the fluid is treated as the same unknown variable in the system, this takes the form of a global mean pressure constraint over the entire domain. Although this approach will increase the problem size, it will only add a single degree of freedom, which will be an additional row and column populated by n_p non-zero entries. Given that the condition number of a system is the ratio of its (modulus of) maximum to minimum eigenvalue, it is possible that this extra equation could give an outlying eigenvalue. If the clustering of eigenvalues for the rest of the system remains the same, the convergence characteristics will be similar, where a single extra iteration of the Krylov iterative method would be required.

The linear elastic equation (3.37) is solved in $\Omega_{t_{n+1}}$ using piecewise quadratic continuous solution space, $\boldsymbol{w}_{n+1} \in W^h(\Omega_{t_{n+1}})^d$ given test space $\boldsymbol{z} \in W^h(\Omega_{t_{n+1}})^d$,

$$\frac{\mu^s}{2} \int_{\Omega_{t_{n+1}}} \mathrm{D}\boldsymbol{w}_{n+1} : \mathrm{D}\boldsymbol{z} \, \mathrm{d}\boldsymbol{x} + \lambda \int_{\Omega_{t_{n+1}}} (\nabla \cdot \boldsymbol{w}_{n+1})(\nabla \cdot \boldsymbol{z}) \, \mathrm{d}\boldsymbol{x} = 0. \tag{3.55}$$

Where the finite element space to find the mesh velocity is defined as,

$$\boldsymbol{z} \in W^h(\Omega_{t_0}) \subset H_0^1(\Omega_{t_0})^d. \tag{3.56}$$

Finally discretising the boundary conditions (3.30) and (3.31), the interface condition becomes,

$$\boldsymbol{w}_{n+1} = \boldsymbol{u}_{n+1}^h \quad \text{on} \quad \Gamma_{t_{n+1}}, \tag{3.57}$$

and boundary conditions,

$$\boldsymbol{w}_{n+1} \cdot \boldsymbol{n} = 0, \ \boldsymbol{n} \cdot \mathrm{D}\boldsymbol{w}_{n+1} \cdot \boldsymbol{t} = 0, \quad \text{on} \quad \partial\Omega_{t_{n+1}}. \tag{3.58}$$

Knowing the discrete ALE velocity $\boldsymbol{w}_{n+1}$, we are able to construct a mapping for $t \in (t_n, t_{n+1}]$,

$$\mathcal{A}_{t_n,t_{n+1}} : \Omega_{t_n} \to \Omega_{t_{n+1}}, \quad \mathcal{A}_{t_n,t_{n+1}}(\boldsymbol{x}_n) = \boldsymbol{x}_n + (t_{n+1} - t_n)\boldsymbol{w}_{n+1}, \tag{3.59}$$

and up to the current timestep the mapping is

$$\mathcal{A}_t = \mathcal{A}_{t_0,t_1}^{-1} \circ \mathcal{A}_{t_1,t_2}^{-1} \ldots \circ \mathcal{A}_{t_n,t_{n+1}}^{-1} \tag{3.60}$$

Two dimensional solid implementation

Since the neo-Hookean model is expressed in the gradient tensor, $\boldsymbol{F}$, solid stress must be expressed as the displacement and velocity within the solid. When using implicit Euler time discretisation, the displacement is simply expressed in velocities as $\boldsymbol{d}_{n+1} = \boldsymbol{d}_n + \Delta t \boldsymbol{u}_{n+1}$. The displacement vector between the undeformed and deformed states of the structure is defined as $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{X}$.

Beginning with the solid stress from equation (3.28), replacing $\boldsymbol{F}\boldsymbol{F}^T = \boldsymbol{B}$ in the left Cauchy-Green tensor, gives

$$\boldsymbol{\sigma}^s = \mu^s J_{\mathcal{L}_t}^{-1}(\boldsymbol{B} - \boldsymbol{I}) - p^s \boldsymbol{I}. \tag{3.61}$$

The Cayley-Hamilton theorem states that every tensor satisfies its own characteristic equation, for example $\det(\lambda \boldsymbol{I} - \boldsymbol{A}) = 0$. For two-dimensions [94], this implies that

$$\boldsymbol{B}^2 - \mathrm{tr}\,(\boldsymbol{B})\,\boldsymbol{B} + \det\,(\boldsymbol{B})\boldsymbol{I} = 0, \tag{3.62}$$

which is related to the Jacobian determinant by $\det\,(\boldsymbol{B}) = \det\,(\boldsymbol{F}\boldsymbol{F}^T) = \det\,(\boldsymbol{F})\det\,(\boldsymbol{F}^T) = J_{\mathcal{L}_t}^2$. The Cayley-Hamilton theorem results in a more complex formulation in three dimensions, which will be discussed later. To find an expression for the Cauchy-Green tensor, Equation (3.62) is multipled by $\boldsymbol{B}^{-1}$ and rearranged to

$$\boldsymbol{B} = \mathrm{tr}\,(\boldsymbol{B})\,\boldsymbol{I} - J_{\mathcal{L}_t}^2 \boldsymbol{B}^{-1}. \tag{3.63}$$

The gradient tensor is expressed in terms of displacement

$$\boldsymbol{F} = \nabla_{\boldsymbol{X}} \boldsymbol{x} = \nabla_{\boldsymbol{X}}(\boldsymbol{X} + \boldsymbol{d}) = \boldsymbol{I} + \boldsymbol{F}\nabla \boldsymbol{d}, \tag{3.64}$$

which can be rearranged as $\boldsymbol{F}^{-1} = (\boldsymbol{I} - \nabla \boldsymbol{d})$, thus

$$\boldsymbol{B}^{-1} = \boldsymbol{F}^{-1}\boldsymbol{F}^{-T} = (\boldsymbol{I} - \nabla \boldsymbol{d})(\boldsymbol{I} - \nabla \boldsymbol{d})^T = \boldsymbol{I} - \mathrm{D}\boldsymbol{d} + \nabla \boldsymbol{d} \nabla^T \boldsymbol{d}. \tag{3.65}$$

Along with the characteristic solution (3.63), these are substituted into the solid stress (3.61),

$$\begin{aligned} \boldsymbol{\sigma}^s &= c_1 J_{\mathcal{L}_t}^{-1}(\boldsymbol{B} - \boldsymbol{I}) - p^s \boldsymbol{I} \\ &= -c_1 J_{\mathcal{L}_t} \boldsymbol{B}^{-1} + c_1 J_{\mathcal{L}_t}^{-1}(\mathrm{tr}\,(\boldsymbol{B}) - 1)\boldsymbol{I} - p^s \boldsymbol{I} \\ &= -c_1 J_{\mathcal{L}_t}(\boldsymbol{I} - \nabla \boldsymbol{d})(\boldsymbol{I} - \nabla \boldsymbol{d})^T + c_1 J_{\mathcal{L}_t}^{-1}(\mathrm{tr}\,(\boldsymbol{B}) - 1)\boldsymbol{I} - p^s \boldsymbol{I} \\ &= c_1 J_{\mathcal{L}_t}(\mathrm{D}\boldsymbol{d} - \nabla \boldsymbol{d} \nabla^T \boldsymbol{d}) + c_1 J_{\mathcal{L}_t}^{-1}(\mathrm{tr}\,(\boldsymbol{B}) - 1)\boldsymbol{I} - p^s \boldsymbol{I}. \end{aligned} \tag{3.66}$$

Further simplifying this and replacing $p' = c_1 J_{\mathcal{L}_t}^{-1}(\mathrm{tr}\,(\boldsymbol{B}) - 1) - c_1 J_{\mathcal{L}_t}$,

$$\boldsymbol{\sigma}^s = c_1 J_{\mathcal{L}_t}(\mathrm{D}\boldsymbol{d} - \nabla \boldsymbol{d} \nabla^T \boldsymbol{d}) + p' \boldsymbol{I} - p^s \boldsymbol{I}, \tag{3.67}$$

as in agreement with [43, 94].

Applying this to the solid term, $\int_{\Omega^s_{t_0}} \frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{F}}(\boldsymbol{F}_{n+1}) : \nabla_{\boldsymbol{X}} \boldsymbol{v}$, in equation (3.52), noting that the hyperelastic solid integral occurs in the initial, undeformed state and updating the displacements as

$$\boldsymbol{d}_{n+1} = \tilde{\boldsymbol{d}}_n + \Delta t \boldsymbol{u}_{n+1}, \tag{3.68}$$

$$\tilde{\boldsymbol{d}}_n = \boldsymbol{d}_n \circ \mathcal{A}^{-1}_{t_n}. \tag{3.69}$$

Referring back to the frame transformation in (3.34), finally the solid term in two-dimensions is computed as

$$\int_{\Omega^s_{t_0}} \frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{F}}(\boldsymbol{F}_{n+1}) : \nabla_{\boldsymbol{X}} \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} = \int_{\Omega^s_{t_{n+1}}} \boldsymbol{\tau}^s : \nabla \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} \tag{3.70}$$

$$= c_1 \int_{\Omega^s_{t_{n+1}}} (\mathrm{D}\boldsymbol{d}_{n+1} - \nabla^T \boldsymbol{d}_{n+1} \nabla \boldsymbol{d}_{n+1}) : \nabla \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} \tag{3.71}$$

$$\begin{aligned} &= \frac{c_1 \Delta t}{2} \int_{\Omega^s_{t_{n+1}}} \mathrm{D}\boldsymbol{u}_{n+1} : \mathrm{D}\boldsymbol{v} \, \mathrm{d}\boldsymbol{x} + \frac{c_1}{2} \int_{\Omega^s_{t_{n+1}}} \mathrm{D}\tilde{\boldsymbol{d}}_n : \mathrm{D}\boldsymbol{v} \, \mathrm{d}\boldsymbol{x} \\ &- \Delta t c_1 \int_{\Omega^s_{t_{n+1}}} (\nabla^T \boldsymbol{u}_{n+1} \nabla \tilde{\boldsymbol{d}}_n + \nabla^T \tilde{\boldsymbol{d}}_n \nabla \boldsymbol{u}_{n+1}) : \nabla \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} \\ &- c_1 \int_{\Omega^s_{t_{n+1}}} \nabla^T \tilde{\boldsymbol{d}}_n \nabla \tilde{\boldsymbol{d}}_n : \nabla \boldsymbol{v} \, \mathrm{d}\boldsymbol{x}. \end{aligned} \tag{3.72}$$

It is worth noting that high order terms $O(\Delta t^2)$ are neglected (from the approximation of $\boldsymbol{d}_{n+1}$) and for an incompressible solid to preserve volume $J_{\mathcal{L}_t}$ is replaced by 1.

Three dimensional solid implementation

For an incompressible three-dimensional neo-Hookean solid,

$$\frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{F}} \boldsymbol{F}^T = (2c_1 - 4c_2 \mathrm{tr}\,(\boldsymbol{B}))\boldsymbol{B} + 4c_2 \boldsymbol{B}^2. \tag{3.73}$$

Again by the Cayley-Hamilton theorem the characteristic equation for a invertible symmetric matrix, expands from (3.62) to,

$$\boldsymbol{B}^3 - \mathrm{tr}\,(\boldsymbol{B})\,\boldsymbol{B}^2 + \gamma \boldsymbol{B} - \det(\boldsymbol{B})\boldsymbol{I} = 0, \tag{3.74}$$

where $\gamma = \frac{1}{2}\left(\operatorname{tr}(\boldsymbol{B})^2 - \operatorname{tr}(\boldsymbol{B}^2)\right)$ and again by incompressibility $\det(\boldsymbol{B}) = 1$.

As a function of displacement, $\boldsymbol{d}$, the three-dimension neo-Hookean stress tensor,

$$\boldsymbol{\sigma}^s = 2c_1(\mathrm{D}\boldsymbol{d} - \nabla\boldsymbol{d}\nabla^T\boldsymbol{d})^2 + 2c_3(x,t)(\mathrm{D}\boldsymbol{d} - \nabla\boldsymbol{d}\nabla^T\boldsymbol{d}) + p''\boldsymbol{I} - p^s\boldsymbol{I}, \tag{3.75}$$

where $p'' = (2c_1\operatorname{tr}(\boldsymbol{B}) - 4c_2\gamma)\boldsymbol{I}$ is unimportant since it will be absorbed into the pressure [43].

The key difference with the two-dimensional implementation is that $c_3(x,t)$ varies with space and time, as a function of the left Cauchy-Green tensor, $\boldsymbol{B}$, as follows [43],

$$c_3 := \frac{c_1}{2}\left(\operatorname{tr}(\boldsymbol{B})^2 - \operatorname{tr}\left(\boldsymbol{B}^2\right) - 4\right) - 4c_2. \tag{3.76}$$

To calculate $\operatorname{tr}(\boldsymbol{B})^2$ and $\operatorname{tr}(\boldsymbol{B}^2)$ the inverse of a matrix of 3 by 3 is required at each point in the solid domain, using the displacement tensor in (3.65).

The three-dimensional of the solid term, which is an extension of (3.72),

$$\begin{aligned}\int_{\Omega^s_{t_0}} \frac{\partial\Psi}{\partial\boldsymbol{F}}(\boldsymbol{F}_{n+1}) : \nabla_{\boldsymbol{X}}\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} = \Delta t \int_{\Omega^s_{t_{n+1}}} \Big(& 2c_1(\mathrm{D}\boldsymbol{u}_{n+1} - \nabla\boldsymbol{u}_{n+1}\nabla^T\tilde{\boldsymbol{d}}_n - \nabla\tilde{\boldsymbol{d}}_n\nabla^T\boldsymbol{u}_{n+1}) \\ & (\mathrm{D}\tilde{\boldsymbol{d}}_n - \nabla\tilde{\boldsymbol{d}}_n - \nabla\tilde{\boldsymbol{d}}_n\nabla^T\tilde{\boldsymbol{d}}_n) \\ & + c_3^n(\mathrm{D}\boldsymbol{u}_{n+1} - \nabla\boldsymbol{u}_{n+1}\nabla^T\tilde{\boldsymbol{d}}_n - \nabla\tilde{\boldsymbol{d}}_n\nabla^T\boldsymbol{u}_{n+1})\Big) : \mathrm{D}\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} \\ + \int_{\Omega^s_{t_{n+1}}} \Big(c_1(\mathrm{D}\tilde{\boldsymbol{d}}_n - \nabla\tilde{\boldsymbol{d}}_n\nabla^T\tilde{\boldsymbol{d}}_n)^2 + c_3^n(\mathrm{D}\tilde{\boldsymbol{d}}_n - \nabla\tilde{\boldsymbol{d}}_n - \nabla\tilde{\boldsymbol{d}}_n\nabla^T\tilde{\boldsymbol{d}}_n\tilde{\boldsymbol{d}}\Big) & : \mathrm{D}\boldsymbol{v}\,\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{3.77}$$

The second integral considers just the updated displacements from the previous time step and will therefore be known at time step $n+1$.

3.6.3 Characteristic-Galerkin method

To approximate the non-linear convection term, the Characteristic-Galerkin approximates the total derivative using a first-order implicit scheme [43], which is easily implementable in FreeFEM. This approach of combining the method of characteristics in the finite element element framework is first analysed by Pironneau [158]. In the Characteristic-Galerkin method, the discrete convection and time derivative terms are combined as a directional derivative along the trajectories of the particles as the total derivative, which is then discretised [158, 159].

Given $t \in (0, T)$ and $\boldsymbol{x} \in \Omega$, let $\mathcal{X}(\tau)$ be a solution of the differential equation at time $\tau < t$,

$$\dot{\mathcal{X}} = \boldsymbol{u}(\mathcal{X}(\tau), \tau) \quad \text{with} \quad \mathcal{X}(t) = x. \tag{3.78}$$

The method of characteristics uses a total derivative, for any given differentiable function for $\alpha : \Omega^t \times (0, T) \mapsto \mathbb{R}^n$,

$$\frac{D\alpha(x,t)}{Dt} = \frac{\partial \alpha}{\partial t} + \boldsymbol{u} \cdot \nabla \alpha. \tag{3.79}$$

If $\boldsymbol{u}$ is the velocity of a fluid, $\mathcal{X}$ is the trajectory of the particle that passes at position x at time t. A first-order approximation for the total derivative of function α using implicit scheme [158],

$$\begin{aligned}\left(\frac{\partial \alpha}{\partial t} + \boldsymbol{u} \cdot \nabla \alpha\right)_{n+1} \approx \frac{1}{k}\Big[&\alpha\left(\mathcal{X}(x, (n+1)k; (n+1)k), (n+1)k\right) \\ &- \alpha\left(\mathcal{X}(x, (n+1)k; nk, nk)\right)\Big],\end{aligned} \tag{3.80}$$

where the time interval $k = \Delta t$, and substituting $\mathcal{X}(x, (n+1)k; (n+1)k) = x$ we arrive at

$$\left(\frac{\partial \alpha}{\partial t} + \boldsymbol{u} \cdot \nabla \alpha\right)_{n+1} \approx \frac{1}{\Delta t}\left(\alpha_{n+1}(x) - \alpha_n\left(\mathcal{X}_n(x)\right)\right), \tag{3.81}$$

where $\mathcal{X}^n$ is an approximation of the interval $\mathcal{X}(x, (n+1)\Delta t; n\Delta t)$. The position of a particle at time t^n which will be at position x at time t_{n+1}, using a first-order approximation, [160],

$$\mathcal{X}(x, (n+1)\Delta t; n\Delta t) \approx \mathcal{X}_n(x) = x - \boldsymbol{u}_n(x)\Delta t. \tag{3.82}$$

Thus a scheme is derived for (3.79) where the differentiable function is velocity, the convection terms becomes,

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \approx \frac{1}{\Delta t}(\boldsymbol{u}_{n+1} - \boldsymbol{u}_n \circ \mathcal{X}_n), \tag{3.83}$$

noting that $\boldsymbol{u}_{n+1}$ and $\boldsymbol{u}_n \circ \boldsymbol{X}_n$ are both defined on Ω_{n+1}. For example discretising the Navier-Stokes equations in time and including this convection becomes,

$$\frac{1}{\Delta t}(\boldsymbol{u}_{n+1} - \boldsymbol{u}_n \circ \mathcal{X}_n) - \nu \Delta \boldsymbol{u}_{n+1} + \frac{1}{\rho}\nabla p_{n+1} = \alpha_{n+1}, \tag{3.84}$$

$$\nabla \cdot \boldsymbol{u}_{n+1} = 0, \tag{3.85}$$

where the following approximation is made $\boldsymbol{u}_n \circ \mathcal{X}_n(x) \approx \boldsymbol{u}_n(x - \boldsymbol{u}_n(x)\Delta t)$. To solve this iteratively it is rearranged to,

$$\frac{1}{\Delta t}\boldsymbol{u}_{n+1} - \nu\Delta\boldsymbol{u}_{n+1} + \frac{1}{\rho}\nabla p_{n+1} = \alpha_{n+1} + \frac{1}{\Delta t}(\boldsymbol{u}_n(x - \boldsymbol{u}_n\Delta t)). \tag{3.86}$$

The key drawback of this method is that it is numerically dissipative [158]. A second-order scheme is also proposed by Pironneau et al. [160]. This approach will generate a linear system of equations at each time step to be solved.

The complete discretised form of the FSI-ALE problem is described by Problem 2, and reads as follows.

Problem 2: Given $\mathcal{A}_{t_n}$ and $\boldsymbol{u}_n^h = \boldsymbol{u}(\mathcal{A}_{t_n}(\boldsymbol{X}), t_n), \forall \boldsymbol{X} \in \Omega_{t_0}$, find $\boldsymbol{u}_{n+1}^h \in V^h(\Omega_{t_{n+1}})^d$, $p_{n+1}^h \in Q^h(\Omega_{t_{n+1}})$ and $\boldsymbol{w}_{n+1}^h \in W^h(\Omega_{t_{n+1}})^d$, such that $\forall \boldsymbol{v} \in V^h(\Omega_{t_{n+1}})^d$, $q \in Q^h(\Omega_{t_{n+1}})$, $\boldsymbol{z} \in W^h(\Omega_{t_{n+1}})^d$, by the following relations

$$\begin{aligned}
&\frac{\rho}{\Delta t}\int_{\Omega_{t_{n+1}}} (\boldsymbol{u}_{n+1}^h - \boldsymbol{u}_n^h \circ \mathcal{X}_n) \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} - \rho \int_{\Omega_{t_{n+1}}} (\boldsymbol{w}_{n+1}^h \cdot \nabla)\boldsymbol{u}_{n+1}^h \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} \\
&+ \frac{\mu^f}{2}\int_{\Omega_{t_{n+1}}^f} \mathrm{D}\boldsymbol{u}_{n+1}^h : \mathrm{D}\boldsymbol{v} \, \mathrm{d}\boldsymbol{x} + \int_{\Omega_{t_0}^s} \frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{F}}(\boldsymbol{F}_{n+1}) : \nabla_{\boldsymbol{X}}\boldsymbol{v} \, \mathrm{d}\boldsymbol{x} + \int_{\Omega_{t_{n+1}}} p_{n+1}^h \nabla \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} \\
&\qquad = \rho \int_{\Omega_{t_{n+1}}} \boldsymbol{g} \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x},
\end{aligned}$$

$$-\int_{\Omega_{t_{n+1}}} q\nabla \cdot \boldsymbol{u}_{n+1}^h \, \mathrm{d}\boldsymbol{x} = 0,$$

$$\frac{\mu^s}{2}\int_{\Omega_{t_{n+1}}} \mathrm{D}\boldsymbol{w}_{n+1}^h : \mathrm{D}\boldsymbol{z} \, \mathrm{d}\boldsymbol{x} + \lambda \int_{\Omega_{t_{n+1}}} (\nabla \cdot \boldsymbol{w}_{n+1}^h)(\nabla \cdot \boldsymbol{z}) \, \mathrm{d}\boldsymbol{x} = 0,$$

where the mesh Ω_{n+1} is updated by (3.43). The solid term is formulated by (3.72) or (3.77), for two- or three-dimensional problems, respectively, where displacements are updated by (3.68) and (3.69). The above FSI system is completed by the Dirichlet and Neumann boundary conditions, (3.15), and the interface condition, (3.14), for the momentum and continuity equations (which are problem-specific and will be defined in later sections), and the interface and boundary conditions, (3.57) and (3.58), for the mesh equation.

From a numerical perspective, the algorithm and solution approach to the above problem is outlined in the next two sections.

3.7 Solution algorithm

Below is an overview of the solution algorithm at each time step using an implicit time discretisation scheme.

```
Algorithm 1: FSI/ALE algorithm solving for each time step.
  Result: Find fluid/solid velocity $\boldsymbol{u}^h_{n+1}$ and pressure $p^h_{n+1}$, and the mesh
          velocity $\boldsymbol{w}^h_{n+1}$ at each time step.
  Set: $t = 0$ and initial flow field $\boldsymbol{u}^h_{t_0}$ in $\Omega_{t_0}$.
    1. $t_{n+1} = t_n + \Delta t$
    2. Solve the FSI equations (3.52) and (3.53) for $\boldsymbol{u}^h_{n+1}$ and $p^h_{n+1}$, given the
       boundary conditions for $\boldsymbol{u}^h_{n+1,D}$.
    3. Solve the linear elastic mesh equation (3.55) for $\boldsymbol{w}^h_{n+1}$, using $\boldsymbol{u}^h_{n+1}$ for the
       interface condition.
    4. Update mesh, $\Omega_{n+1} = \Omega_n + \Delta t \boldsymbol{w}^h_{n+1}$.
```

The solution approach for the FSI system, as outlined in Algorithm 1, in Step 2 and the mesh equation in Step 3 can be a direct or iterative solver, as this thesis will explore and discuss in future chapters.

3.8 Discrete saddle-point system

Using mixed finite elements, the discrete ALE formulation will form a system of equations with the following block structure,

$$\begin{bmatrix} \boldsymbol{A} & B^T \\ B & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u}_{n+1} \\ p_{n+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F} \\ 0 \end{pmatrix}, \tag{3.87}$$

with each block a discrete form an operator acting upon unknowns $\boldsymbol{u}_{n+1}$ and p_{n+1}.

By combining equations (3.52) and (3.72), with the convection term from the characteristic Galerkin method in (3.83), we will arrive at the first set of equations of the saddle-point system, in two dimensions,

$$
\begin{aligned}
&\frac{\rho}{\Delta t}\int_{\Omega_{t_{n+1}}}(\boldsymbol{u}^h_{n+1}-\boldsymbol{u}^h_n\circ\mathcal{X}_n)\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x}-\rho\int_{\Omega_{t_{n+1}}}(\boldsymbol{w}^h_{n+1}\cdot\nabla)\boldsymbol{u}^h_{n+1}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x}\\
&+\frac{\mu^f}{2}\int_{\Omega^f_{t_{n+1}}}\mathrm{D}\boldsymbol{u}^h_{n+1}:\mathrm{D}\boldsymbol{v}\mathrm{d}\boldsymbol{x}-\Delta t c_1\int_{\Omega^s_{t_{n+1}}}(\nabla^T\boldsymbol{u}^h_{n+1}\nabla\tilde{\boldsymbol{d}}_n+\nabla^T\tilde{\boldsymbol{d}}_n\nabla\boldsymbol{u}^h_{n+1}):\nabla\boldsymbol{v}\mathrm{d}\boldsymbol{x}+\int_{\Omega_{t_{n+1}}}p^h_{n+1}\nabla\cdot\boldsymbol{v}\mathrm{d}\boldsymbol{x}\\
&\quad=\rho\int_{\Omega_{t_{n+1}}}\boldsymbol{g}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x}-\frac{c_1}{2}\int_{\Omega^s_{t_{n+1}}}\mathrm{D}\tilde{\boldsymbol{d}}_n:\mathrm{D}\boldsymbol{v}\,\mathrm{d}\boldsymbol{x}+c_1\int_{\Omega^s_{t_{n+1}}}\nabla^T\tilde{\boldsymbol{d}}_n\nabla\tilde{\boldsymbol{d}}_n:\nabla\boldsymbol{v}\,\mathrm{d}\boldsymbol{x}
\end{aligned}
\tag{3.88}
$$

The second set of equations is just the continuity equation, (3.53), again stated for completeness

$$
-\int_{\Omega_{t_{n+1}}}q\nabla\cdot\boldsymbol{u}_{n+1}\,\mathrm{d}\boldsymbol{x}=0. \tag{3.89}
$$

By applying the finite element approximations in (3.50) and (3.51) we arrive at (3.88), given in a fully discrete block form

$$
\left(\frac{1}{\Delta t}\mathbf{M}+\mathbf{K}+\Delta t\mathbf{K}^s-\Delta t\mathbf{D}^s\right)\boldsymbol{u}_{n+1}-\mathbf{B}^T p_{n+1}=\frac{1}{\Delta t}\mathbf{C}-\mathbf{N}-\mathbf{E}^s \tag{3.90}
$$

and the mass continuity equation (3.53) is

$$
-\mathbf{B}\boldsymbol{u}_{n+1}=\mathbf{0}. \tag{3.91}
$$

The contributions to this system are as follows

$$
\mathrm{M}_{ij}=\rho(\phi_i\cdot\phi_j), \tag{3.92}
$$

$$
\mathrm{K}_{ij}=\frac{\mu^f}{2}(\mathrm{D}\phi_i:\mathrm{D}\phi_j), \tag{3.93}
$$

$$
\mathrm{K}^s_{ij}=\frac{c_1}{2}(\mathrm{D}\phi_i:\mathrm{D}\phi_j), \tag{3.94}
$$

$$
\mathrm{D}^s_{ij}=c_1(\nabla\phi_i\nabla\tilde{d}_j+\nabla\tilde{d}_i\nabla\phi_j):\nabla\phi_j, \tag{3.95}
$$

$$
\mathrm{B}_{ij}=\psi_i\nabla\cdot\phi_j, \tag{3.96}
$$

$$
\mathrm{C}_j=\rho u^{\mathrm{old}}_j\phi_j, \tag{3.97}
$$

$$
\mathrm{N}_j=\rho(w_j\cdot\nabla)\phi_j, \tag{3.98}
$$

$$
\mathrm{E}^s_j=\frac{c_1}{2}(\mathrm{D}\tilde{d}_j:\mathrm{D}\phi_j)+c_1(\nabla\tilde{d}_i\nabla\tilde{d}_j:\nabla\phi_j). \tag{3.99}
$$

As before ϕ denotes the basis function for velocity, and ψ for pressure. Matrix **M** is known as the mass matrix, **K** corresponds to the discretised Laplace operator. The $u_j^{\text{old}} = \boldsymbol{u}_n(x - \boldsymbol{u}_n \Delta t)$ velocity in term (3.97) arises from the approximation made in the Characteristic-Galerkin method.

The above case is the discrete form the of the two-dimensional solid, for comparison the three-dimensional solid is given,

$$\begin{aligned} \mathrm{D}_{ij}^s &= 2c_1(\mathrm{D}\phi_j - \nabla\phi_j\nabla\tilde{d}_i - \nabla\tilde{d}_j\nabla\phi_i)(\mathrm{D}\tilde{d}_j - \nabla\tilde{d}_j - \nabla\tilde{d}_i\nabla\tilde{d}_j) \\ &\quad + c_3(\mathrm{D}u_j - \nabla u_j\nabla\tilde{d}_i - \tilde{d}_j\nabla u_i) : \mathrm{D}\phi_j, \end{aligned} \tag{3.100}$$

$$\mathrm{E}_j^s = c_1(\mathrm{D}\tilde{d}_j - \nabla\tilde{d}_j\nabla\tilde{d}_i)^2 + c_3(\mathrm{D}\tilde{d}_j - \nabla\tilde{d}_j - \nabla\tilde{d}_j\nabla\tilde{d}_i) : \mathrm{D}\phi_j. \tag{3.101}$$

Thus comparing these contributions, (3.93)-(3.99), as well as (3.100) and (3.101), with the block system in (3.87), the **A** block are the terms involving the velocity variables, collects the time derivative, advection and diffusion operators, and solid contributions,

$$\mathbf{A} = \frac{1}{\Delta t}\mathbf{M} + \mathbf{K} + \Delta t\mathbf{K}^s - \Delta t\mathbf{D}^s, \tag{3.102}$$

Comparing this to a purely fluid problem [66], the LHS of the system now includes addition contributions from the stiffness term $\mathbf{K}^s$ and $\mathbf{D}^s$ term, both arising from the discrete hyperelastic solid. The rectangular block denoting the discrete gradient operator remains the same for both a fluid and a FSI problem. All other terms are moved to the RHS of equation (3.90) contributing to the vector **F** in the system,

$$\mathbf{F} = \frac{1}{\Delta t}\mathbf{M}\boldsymbol{u}_n + \mathbf{N}\boldsymbol{u}_n - \mathbf{N}\boldsymbol{w}_n - \mathbf{E}^s\tilde{\boldsymbol{d}}_n. \tag{3.103}$$

3.9 Concluding remarks

This chapter outlines the ALE formulation for the FSI, applying an incompressible Newtonian fluid and an incompressible neo-Hookean hyperelastic solid. The weak formulation of the ALE description is defined, which is then discretised in both space and time. To linearise the convection term, a Characteristic-Galerkin approach is used to approximate the total derivative in the FSI equations. A mixed-order finite element discretisation in space is applied, using a different test space for velocity and pressure, which will eventually lead to a discrete saddle point structure of the system.

As with this interface fitted approach, the mesh movement is treated using a linear elastic equation. This is discretised to formulate a separate system from the FSI equations, which will be solved sequentially. It is coupled to the FSI equation by using the velocity at the fluid-structure interface. Although adaptive mesh refinement and discrete remeshing will permit larger solid deformation, by maintaining mesh quality, this will not be explored in this thesis.

Although this discretisation approach for the incompressible neo-Hookean solid is simple in its approach, as a result of the Cayley-Hamilton theorem having a different characteristic equation between three and two dimensions, the three-dimensional form does require additional terms, as demonstrated in Section 3.6.2.

Chapter 4

2D finite element spaces

As discussed previously, the most common finite element space for incompressible flow problems is the Taylor-Hood pairing, such as P_2/P_1 (which considers a continuous piecewise quadratic approximation for velocity and a continuous piecewise linear approximation for pressure) is the lowest order stable choice. A continuous pressure space is generally suitable within a continuum such as in a fluid alone. When a discontinuity is considered across a fluid-solid interface, there may be a genuine discontinuity in pressure, and it can be a challenge to resolve accurately with continuous elements. To avoid the computational cost of resolving the near-interface region in very fine detail when using a continuous pressure space, a more suitable discontinuous pressure finite element space can be considered.

This chapter discusses the importance of using appropriate finite element spaces when solving FSI problems within the framework of an ALE method, in particular accurately capturing the pressure solution at the fluid-solid interface to accurately determine the deflection of the solid. To demonstrate this, we introduce and test three common mixed velocity-pressure finite element spaces using two different geometries, presenting both a steady-state and periodic solid response. Throughout this chapter, the discussion will consider two-dimensional triangular elements.

4.1 Mixed order spaces

To describe finite element spaces, the conventional notation for triangles and tetrahedrons is P_m/P_n, which means that the velocity space is approximated with a continuous piecewise polynomial of degree m for each component of the velocity. The pressure is approximated by a continuous piecewise polynomial of degree n, with the exception when $n = 0$ is used where a piecewise discontinuous constant pressure approximation is used.

The first point of discussion is around using either an equal- or mixed-order approximation of the finite element space. This refers to the order of the basis function when approximating both the velocity and pressure; equal order refers to using the same order approximation for both fields, whereas mixed order typically uses a higher-order piecewise polynomial approximation for velocity relative to pressure. Taking this in relation to the governing equation, we note that velocity involves a higher-order operator, $\nabla^2 \boldsymbol{u}$, than pressure, ∇p.

Thus we arrive at the following velocity-pressure space pairing (for triangles), P_m/P_{m-1}, which is stable when $m \geq 2$. These ubiquitous elements known as Taylor-Hood were proven to be stable for discretisation of incompressible flow by Taylor and Hood [185].

When discussing the stability of the different element spaces, we refer to the Ladyzhenskaya-Babuka-Brezzi (LBB) stability condition (also referred to as *div-stability* or *inf-sup* condition), first coined by Oden and Kikuchi [149] from a collection of separate works [6, 32, 123].

An equal order approximation is possible but generally can only give 'stability' if the mass continuity equation is modified by weakening it, for example $\nabla \cdot \boldsymbol{u} = \varepsilon f(p)$, where the small value of ε acts as a stabilisation [87, Ch. 3.13.3]. Equal-order approximation may give unphysical results unless some form of special treatment is applied, such as a bubble function or a Bercovier-Pironneau element (linear velocity functions on a half space, sometimes referred to as $P_1 iso P_2$).

Mixed finite elements, which do offer good robustness, discretise to form a saddle-point system. The discrete saddle point system is indefinite, having both negative and positive eigenvalues [3]. This rules out the use of certain direct methods (for example, Cholesky factorisation) and some iterative methods (for example, conjugate gradient).

To demonstrate the *inf-sup* condition, the standard incompressible Stokes problem is considered. This may be expressed in the well-defined bilinear form

$$a(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) = \langle \boldsymbol{f}, \boldsymbol{v} \rangle \quad \forall\, \boldsymbol{v} \in V, \tag{4.1a}$$

$$b(\boldsymbol{u}, q) = \langle g, q \rangle \quad \forall\, q \in Q, \tag{4.1b}$$

where the Hilbert spaces are defined as $V = H_0^1(\Omega)^d$ and $Q = L_0^2(\Omega)$. This continuous representation seeks the unknowns $(\boldsymbol{u}, p) \in V \times Q$, with the given $(\boldsymbol{f}, g) \in V' \times Q'$ (where V' and Q' are the dual spaces). The mixed finite element discretisation discussed above can be written in the archetypal block structure of the saddle-point system,

$$\begin{bmatrix} \boldsymbol{A} & B^T \\ B & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ g \end{pmatrix}. \tag{4.2}$$

Introducing the linear operators mapping from the original to the dual spaces,

$$\boldsymbol{A} : V \mapsto V', \quad \langle \boldsymbol{A}\boldsymbol{u}, \boldsymbol{v} \rangle = a(\boldsymbol{u}, \boldsymbol{v}), \tag{4.3a}$$

$$B : V \mapsto Q', \quad \langle B\boldsymbol{v}, q \rangle = b(\boldsymbol{v}, q), \tag{4.3b}$$

$$B^T : Q \mapsto V' \quad \langle \boldsymbol{v}, B^T q \rangle = b(\boldsymbol{v}, q). \tag{4.3c}$$

Letting $a : V \times V \mapsto \mathbb{R}$ and $b : V \times Q \mapsto \mathbb{R}$ be the bilinear forms. It is assumed that the bilinear forms are continuous,

$$a(\boldsymbol{u}, \boldsymbol{v}) \leq C \left\| \boldsymbol{u} \right\| \left\| \boldsymbol{v} \right\|, \quad \forall\, \boldsymbol{u}, \boldsymbol{v} \in V, \tag{4.4a}$$

$$b(\boldsymbol{v}, q) \leq C \left\| \boldsymbol{v} \right\| \left\| q \right\|, \quad \forall\, \boldsymbol{v} \in V, q \in Q, \tag{4.4b}$$

and are bounded, such that there exists a positive real number C. It may be proved that the following inf-sup condition for B,

$$\inf_{q \in Q} \sup_{\boldsymbol{v} \in V} \frac{b(\boldsymbol{v}, q)}{\left\| \boldsymbol{v} \right\| \left\| q \right\|} = \beta > 0, \tag{4.5}$$

implies that there exists a unique solution pair. The complete derivation is given by Brenner and Scott [31], where the conditions above are satisfied for P_m/P_{m-1} under $m \geq 2$.

4.2 Finite element spaces

Three elements types have been identified for investigation within the ALE framework to demonstrate the effect of capturing an accurate pressure solution. In each case, the velocity space is kept at the lowest order to remain LBB stable, without using more "complex" element types in order to ensure the computational efficiency is prioritised.

For the following approximation, $\mathcal{T}_h$ is the triangulation (for triangles in two dimensions and tetrahedrons in three dimensions), with each element denoted as k. The velocity space is represented by the piecewise-continuous quadratic function in all the cases considered in this chapter. The piecewise quadratic continuous finite element space is defined,

$$V_h^2 = \{\boldsymbol{v} : \boldsymbol{v} \in H^1(\Omega),\, \forall k \in \mathcal{T}_h : \boldsymbol{v}_k \in P_2(k)\}, \tag{4.6}$$

where P_2 is a set of polynomials of degree ≤ 2.

4.2.1 P2/P1

As discussed before, the simplest LBB-stable element is the Taylor-Hood element when $m = 2$. It is worth noting that any lower (e.g. $m = 1$) does not meet the LBB stability condition. This ensures a unique and stable solution that can convergence at an optimal rate.

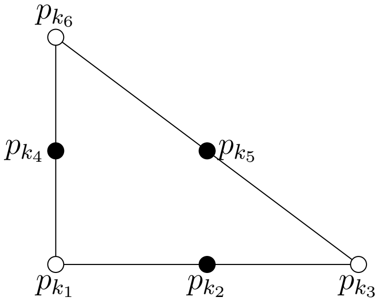


Figure 4.1: Taylor-Hood, P_2/P_1 degrees of freedom on a 2D triangle; ● velocity, ○ velocity and continuous pressure.

The piecewise linear continuous finite element space for pressure is,

$$Q_h^1 = \{q : q \in H^1(\Omega),\ \forall k \in \mathcal{T}_h : q_k \in P_1(k)\}, \tag{4.7}$$

where P_1 is the set of polynomial of degree one.

Taylor-Hood elements only impose the requirement that the mass continuity, $\nabla \cdot \boldsymbol{u} = 0$, is enforced globally in a weak sense. The continuous pressure space results in one of the drawbacks of this type of element, a lack of element-wise mass conservation. A single-pressure mode is present with the Taylor-Hood element, thus the pressure is defined up to a constant. Typically, a single pressure degree of freedom can be restricted at a boundary, stopping the pressure "floating". An alternative is to impose a mean pressure constraint on the whole of the pressure fields. To ensure consistency between all pressure spaces, a mean pressure constraint approach is used, as described in (3.54).

4.2.2 P2/P0

The removal of an order from the pressure space will no longer capture the linear variation on each element, and the P_0 pressure is now piecewise discontinuous. This provides the benefit of being able to naturally capture discontinuities with the pressure solution, which is particularly useful for problems involving discontinuous mediums, such as a fluid-solid interface.

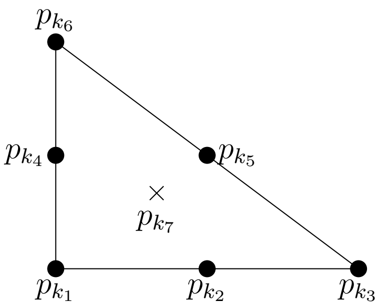


Figure 4.2: P_2/P_0 degrees of freedom on 2D triangle; ● velocity, × discontinuous pressure.

The discontinuous finite element space is defined as,

$$Q_h^0 = \{q : q \in L^2(\Omega),\ \forall k \in \mathcal{T}_h : q_k = \alpha_k\}, \tag{4.8}$$

where $\alpha_k \in \mathbb{R}$ is a real constant.

One key disadvantage of this finite element pairing is that it is only first-order accurate [87]. Again, P_2/P_0 has a single hydrostatic mode and therefore can be constrained by the mean pressure. It is worth noting that this is the lowest order, LBB stable, discontinuous velocity-pressure pairing, noting that P_1/P_0 is not LBB stable [87].

The discontinuous pressure across each element, ensures a local element-wise mass balance [87].

4.2.3 P2/(P1+P0)

Due to the continuous nature of the linear pressure approximation in the standard P_2/P_1 Taylor-Hood formulation, one disadvantage is not being able to maintain element-wise mass conservation [87]. Numerical tests by Tidd et al. [188] find that a lack of local mass conservation can lead to unphysical results, with enrichment giving conservation for each element instead of over the entire domain. To circumvent this, the linear approximation of the pressure space is augmented by an element-wise discontinuous constant, essentially "enriching" the pressure approximation. Element-wise mass conservation arises from this additional test function, $q_h = 1$, equal to one within the element of interest and being zero in all other elements. Thus for mass conservation approximation, $\int_\Omega q_h \nabla \cdot \boldsymbol{u}_h = 0$, and the divergence theorem,

$$\int_{\Omega_e} \nabla \cdot \boldsymbol{u}_h = \int_{\partial\Omega_e} \boldsymbol{u}_h \cdot \boldsymbol{n} = 0, \tag{4.9}$$

holds true over every element.

Using stability analysis Thatcher and Silvester [187] originally show that this element, supplementing the continuous linear pressure space, is stable for a wide range of triangular grids. Boffi et al. [28] generalise this to cover Taylor-Hood elements for triangular and tetrahedral elements, under the condition that every element has at least one vertex in the interior of the domain. Boffi et al. [27] test this augmented Taylor-Hood element type with an immersed boundary method in two dimensions, demonstrating better mass conservation and convergence rates.

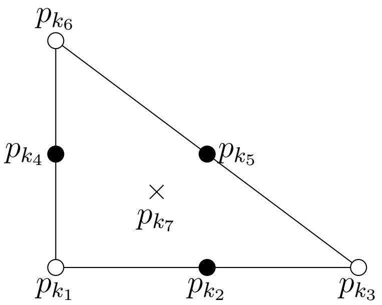


Figure 4.3: $P_2/(P_1 + P_0)$ degrees of freedom on 2D triangle; ● velocity, ○ velocity and continuous pressure and × discontinuous pressure.

For the linear, piecewise discontinuous function the space is defined as,

$$Q_h^{(1+0)} = \{q : q \in L^2(\Omega), \forall k \in \mathfrak{T}_h : q_k = q_{1,k} + q_{0,k}, q_{1,k} \in P_1(k), q_{0,k} = \alpha_k\}, \tag{4.10}$$

as before notating the polynomial of degree one over the element, with the additional element-wise discontinuous constant.

This choice of element pair is LBB stable, however one of the key disadvantages of these triangular elements is the presence of two hydrostatic modes (one for each component of the pressure space) [87]. Since the total pressure is the sum of the linear and constant approximation (P_1 and P_0), the mean pressure is constrained in the same sense, as done in equation (3.54),

$$\int_\Omega p^{P_1} + p^{P_0} \, \mathrm{d}\boldsymbol{x} = 0. \tag{4.11}$$

Thus, for this pressure space, the mean pressure constraint is enforced as the total mean pressure. This is instead of a mean pressure constraint for each component, since the global constant can be represented exactly by the means of each pressure component [28].

4.3 Numerical experiments

The following numerical tests are designed to investigate the relative performance of using discontinuous pressure finite element spaces (compared to P_2/P_1) for fluid-structure interaction problems. Specifically, we consider whether there is an accuracy or efficiency benefit in resolving the pressure at the interface using a discontinuous approximation rather than a continuous one. As we will see, there is typically a jump in pressure across the interface, and so the P_1 elements will smooth this out across the neighbouring elements.

4.3.1 2D Test 1: Filament in a cavity flow

This case considers an enclosed domain with a deformable solid filament fixed to the bottom. The fluid occupies the region $\Omega^f \cup \Omega^s = [0,2] \times [0,1]$, and the solid filament occupies the region $\Omega^s = [0.97, 1.03] \times [0, 0.8]$ when undeformed. The fluid is driven by a shearing lid applied horizontally to the upper boundary. This geometric setup was first proposed by Baaijens [5], and then examined again using a fictitious domain approach [114, 209].

The material properties for the fluid-solid system and the linear elastic-solid model governing the response of the mesh are given in Table 4.1. Note that the density ratio between the solid and fluid $\rho^f = \rho^f/\rho^s = 1$, which eliminates any buoyancy effects. The linear elastic mesh model has a Poisson's ratio of $\nu^m = 0.25$, providing a sufficiently stiff response to the deforming solid filament, which for two dimensions is sufficient to maintain mesh cell quality.

Fluid-solid parameters		
Solid density	ρ^s	100
Fluid density	ρ^f	100
Solid constant	c_1	2×10^6
Fluid viscosity	μ^f	10

Mesh parameters		
First Lamé parameter	λ^m	10
Second Lamé parameter	μ^m	10

Table 4.1: Material properties of the fluid-solid system (left) and the elastic solid governing the mesh deformation (right).

The material properties outlined in Table 4.1 are applied to the discrete FSI system in (3.88) and to the linear elastic mesh equation in (3.55). The densities occupy the regions $\rho = \rho^f \in \Omega^f$ and $\rho = \rho^s \in \Omega^s$.

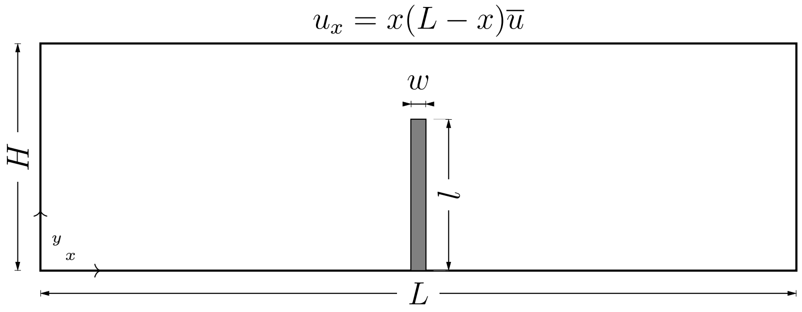


Figure 4.4: Two dimensional geometry of filament in a shearing lid flow, deformable solid highlighted in the region of grey.

The simulation is begun from an undeformed state, as shown in Figure 4.4, and from an initial state of zero velocity in the domain. The flow is initiated by applying a horizontal velocity to the upper boundary, given by equation (4.12). Although somewhat unphysical, the quadratic profile ensures that the velocities at the upper corners of the domain are zero, ensuring conformity with the impermeability condition (zero normal velocity) applied to the vertical boundaries. The x-component of the lid velocity is prescribed as,

$$u_x(x,t) = \begin{cases} x(L-x)\overline{u}\left(\frac{(1-\cos(2\pi t))}{2}\right) & t < 0.5 \\ x(L-x)\overline{u} & t \geq 0.5, \end{cases} \tag{4.12}$$

where using a cosine profile permits a continuous increase in velocity avoiding an initial impulse from its stationary starting state. The velocity scale is set to $\overline{u} = 1$. The steady increase in applied velocity also helps to maintain the global incompressibility condition $\nabla \cdot \boldsymbol{u} = 0$, which can be violated in the case of an impulsive boundary condition. No-slip and impermeable ($\boldsymbol{u} = 0$) boundary conditions are applied to both sides of the domain, as well as the bottom boundary on either side of the filament. Time stepping is conducted with a simple backward Euler scheme, with a constant time step of $\Delta t = 0.001$s.

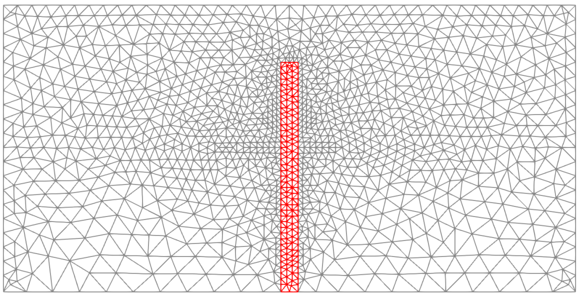


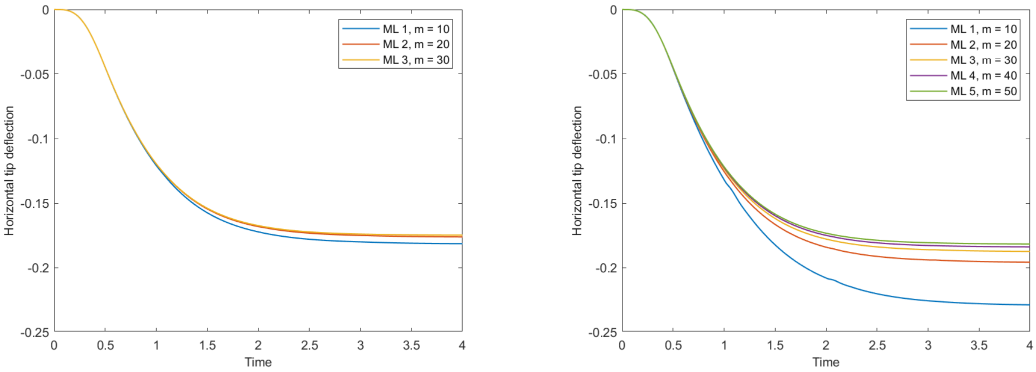
Figure 4.5: Example mesh for the filament in shearing lid case, mesh $m = 10$, solid region coloured in red.

The dimensions of the problem are identical to the original setup, with the geometric properties shown in Table 4.2. An example mesh is shown in Figure 4.5, the resolution is characterised by a parameter, m, controlling the number of cell divisions along each boundary, on the vertical boundaries $m_y = 2m$ and $m_x = 4m$ along the top horizontal boundary. The mesh is constrained to exactly capture the fluid-solid interface. The mesh is modified to include points near the corners of the fluid domain to ensure that any single element does not have all vertices on a boundary. The two-dimensional triangular mesh is generated using FreeFEM's in-built mesh generator.

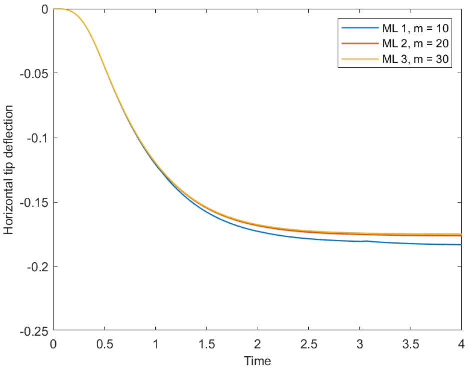
Parameter		Value
Channel length	L	2.0
Channel height	H	1.0
Filament width	w	0.06
Filament height	l	0.8

Table 4.2: Geometric parameters for Figure 4.4.

The simulations are run from an undeformed state until the horizontal tip deflection of the filament has reached a constant steady-state deflection, as shown in Figure 4.6. Finite element pairings P_2/P_0, P_2/P_1 and $P_2/(P_1 + P_0)$ are compared using several successively finer meshes. To examine convergence with mesh resolution, the horizontal deflection of the tip of the filament, located at $(1.0, 0.8)$ in the reference configuration, is measured in time.



(a) P_2/P_0 (b) P_2/P_1



(c) $P_2/(P_1 + P_0)$

Figure 4.6: Horizontal tip deflection in the two-dimensional shear driven cavity for a given mesh resolution.

The discontinuous pressure space of P_2/P_0 is capable of convergent to a mesh-independent solution with a coarser mesh compared to P_2/P_1. Furthermore, each of these sequences of approximation appears to converge to different mesh-independent steady-state deflections. Specifically, both P_2/P_0 and $P_2/(P_1 + P_0)$, with discontinuous pressure spaces, convergence to a deflection of ydis $= -0.175$, while P_2/P_1 convergences to ydis $= -0.182$. A difference of 3.85% between the two deflections.

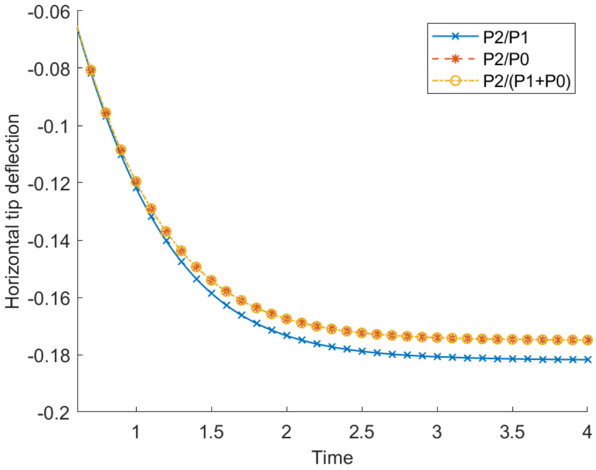


Figure 4.7: Comparison of mesh independent horizontal tip deflections; P_2/P_0, mesh level 3, P_2/P_1, mesh level 5, and $P_2/(P_1 + P_0)$, mesh level 3.

In Figure 4.7 the deflection of the filament tip is compared across the three pressure spaces using mesh converged solutions, as demonstrated in Figures 4.6. For this specific case, there is a very close agreement between P_2/P_0 and $P_2/(P_1+P_0)$ solutions, suggesting that the discontinuous pressure approximation is the dominant feature in accurately capturing the deflection of the solid.

The difference between sequential levels of mesh refinement and solid tip deflection is shown in Table 4.3. The number of degrees of freedom is given for the velocity and pressure spaces used. This is further evidence that for the P_2/P_1 pressure space even at the finest mesh resolution, $m = 50$, the difference in tip deflection compared to the previous resolution is still large, and the solution is far from mesh-converged. Overall, the continuous pressure space requires a higher mesh resolution relative to the discontinuous pressure spaces to reach a similar degree of mesh convergence because it cannot represent a genuine discontinuity at the interface. By extrapolating the convergence rate of the P_2/P_1 pressure space, to reach a similar order of percentage of deflection difference ($\sim 0.7\%$) to the other pressure spaces, will require > 400000 degrees of freedom, approximately four times the number of degrees of freedom relative to the other pairs of finite elements.

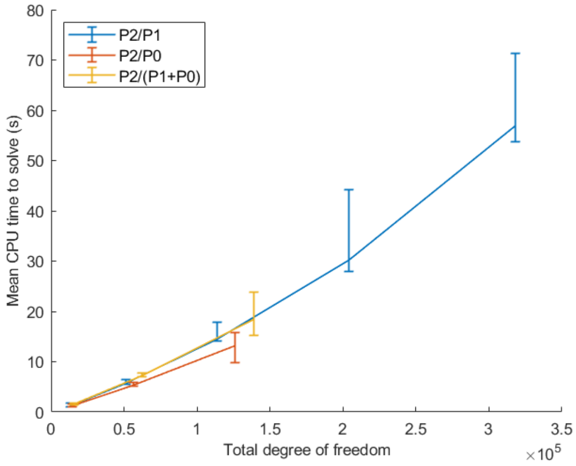


Figure 4.8: Mean solution time using the UMFPACK direct solver [48], bars show the fluctuations about the mean of the time to solve at each given time step.

Although the accuracy of discontinuous pressure approximations is better than the continuous approximation, the solution times using a sparse direct solver for P_2/P_0 are shown to be faster compared to the other pressure spaces, as shown in Figure 4.8. Discontinuous pressure spaces, P_2/P_1 and $P_2/(P_1 + P_0)$, perform almost identically when measuring the time to solve, when comparing the case which has fewer degrees of freedom.

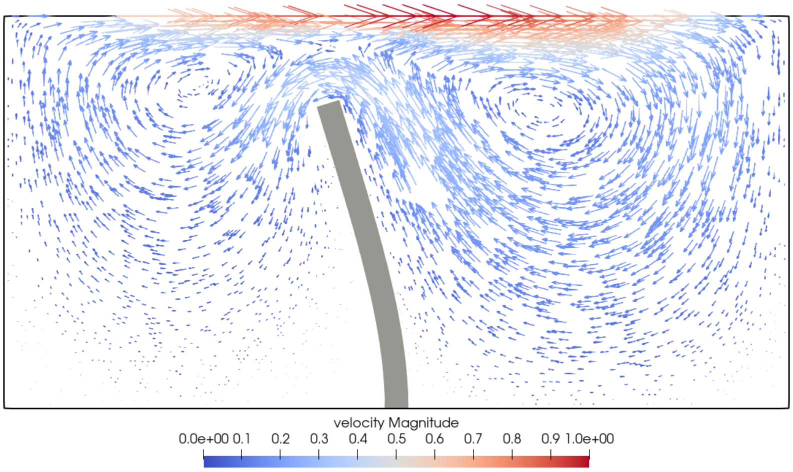
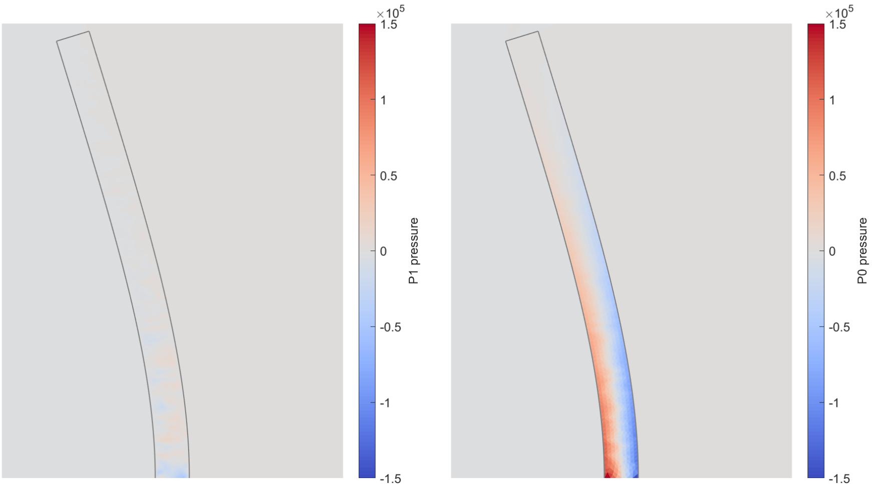


Figure 4.9: Velocity vector plot, P_2/P_0, $m = 30$, $t = 3.99$. The solid region shaded in grey.

Figure 4.9 shows a velocity vector plot, at the final steady-state deflection of the solid. Although the velocity is imposed in a positive x direction at the lid, the recirculation of the fluid in the sub-cavities bound by the deforming solid and the fluid domain causes the filament to deflect in the opposing direction.

Figure 4.9 is rendered using open source scientific visualisation software ParaView. It can either visualise continuous linearly varying or constant discontinuous functions, however, not together. Since ParaView does not have the functionality to plot the combination of linear and element-wise discontinuous functions, it is therefore a poor tool for visualising the enriched pressure spaces. One approach could be to visualise the total pressure by interpolating either onto the P_1 or P_0 spaces on a much finer mesh. Alternatively, we choose to use MATLAB is used to plot all pressure spaces, as shown in Figures 4.10.

The three pressure plots in Figures 4.10, capture the pressure distribution in the final time step using their respective "mesh independent" solution. The largest extremes in pressure are observed at the interface of the solid near the bottom boundary of the fluid domain. With the continuous pressure space, P_2/P_1, it is noticeable that the extreme peak of the pressure is offset from the fluid-solid interface, which runs parallel to the interface along the solid filament. We can clearly see the pressure distribution captured by the P_0 and $P_1 + P_0$ pressure spaces.



(a) P_1 (b) P_0

Figure 4.11: Pressure field plots decomposed into both the piecewise linear continuous and piecewise constant discontinuous contributions. For the same region, time step and mesh resolution as shown in Figure 4.10c.

To better understand the performance of $P_2/(P_1 + P_0)$, the pressure contributions are decomposed into P_1 and P_0 and compared in Figure 4.11. Most of the pressure appears to come from the piecewise constant, discontinuous contribution. There is a small contribution from the P_1 space in the bottom region of the filament, near the boundary of the domain.

Mesh Level	m	Degrees of Freedom			% difference
		Velocity	Pressure	Total	
1	10	11458	2812	14270	-
2	20	45358	11236	56594	-2.8905
3	30	101170	25138	126308	-0.77006

(a) P_2/P_0

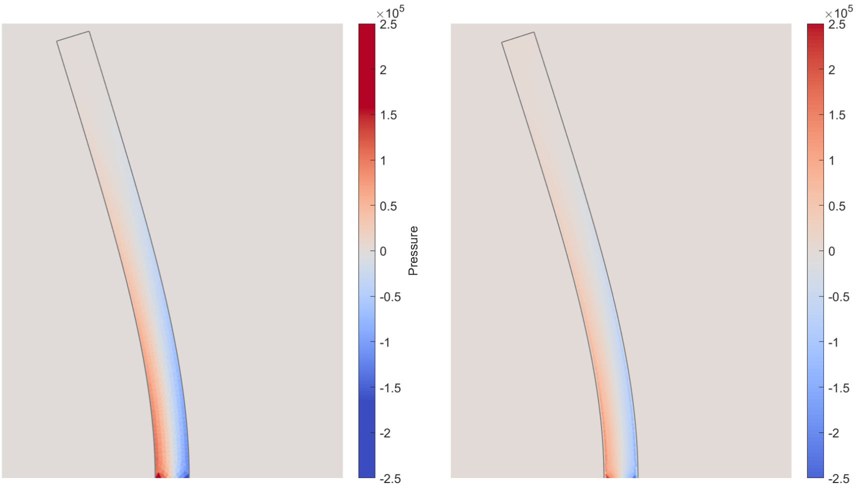
Mesh Level	m	Degrees of Freedom			% difference
		Velocity	Pressure	Total	
1	10	11458	1459	12917	-
2	20	45358	5722	51080	-14.4474
3	30	101170	12724	113894	-4.2597
4	40	181206	22754	203960	-1.8845
5	50	282866	35487	318353	-1.1992

(b) P_2/P_1

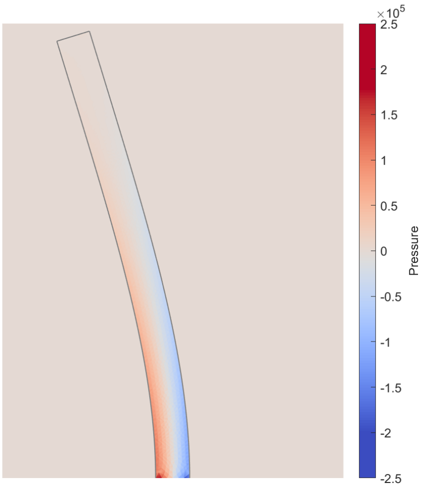
Mesh Level	m	Degrees of Freedom			% difference
		Velocity	Pressure	Total	
1	10	11458	4271	15729	-
2	20	45358	16958	62316	-3.8222
3	30	101170	37862	139032	-0.69158

(c) $P_2/(P_1 + P_0)$

Table 4.3: Degrees of freedom, with percentage difference in horizontal tip deflection between subsequent mesh refinements.



(a) P_2/P_0, $m = 30$, $t = 3.99$ (b) P_2/P_1, $m = 50$, $t = 3.99$

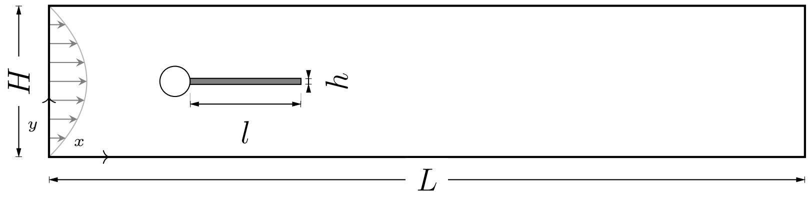


(c) $P_2/(P_1 + P_0)$, $m = 30$, $t = 3.99$

Figure 4.10: Pressure contour, around the solid filament. The region show is bound by $x = [0.7, 1.3]$ and $y = [0, 0.8]$.

4.3.2 2D Test 2: Filament behind a cylinder

This case was designed as an FSI benchmark, first proposed by Turek and Hron [192], and then further compared with several methods in Turek et al. [193]. The setup considers a pipe flow with a deformable filament attached behind a rigid solid cylinder.



(a) Fluid channel with solid.

(b) Solid geometry in the initial configuration.

Figure 4.12: Two dimensional geometry of filament behind a cylinder in a pipe flow, from the FSI benchmark proposed by Turek and Hron [192]. The deformable solid filament is highlighted in grey, and the rigid cylinder is shown filled in white.

The setup of the geometry with the filament attached to the downstream side of the fixed cylinder is shown in Figure 4.12, with the corresponding geometric dimensions in Table 4.4. The fluid domain occupies a region such that $\Omega^f \cup \Omega^s = [0, 2.5] \times [0, 0.41]$, and the solid domain $\Omega^s = [0.25, 0.60] \times [0.195, 0.215]$. For this problem, the time step is implemented again using backward Euler with $\Delta t = 0.005$, $t > 0$. The location of the centre of the fixed cylinder and the horizontal centerline of the filament (in the initial undeformed state) is at $y^s = 0.200$, so it is placed just below the horizontal centerline of the fluid channel, which is at $y^f = 0.205$. This geometric asymmetry initiates the unsteady flow and the oscillation of the filament, without the need for any other external stimuli. A reference point is located at the tip of the solid filament to measure its displacement, shown in Figure 4.12b.

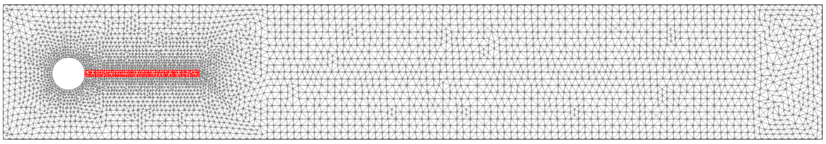
The material properties used in these simulations are shown in Table 4.5. The fluid-solid material properties are chosen to match the benchmark cases from the literature [192]. For this setup, the linear elastic mesh equation is selected to have a much stiffer mesh response relative to the previous case, with a Poisson's ratio of $\nu^m = 0.4545$, to maintain mesh quality in the case where the density of elements around the tip of the filament is high.

A parabolic inlet velocity profile is applied at the inlet

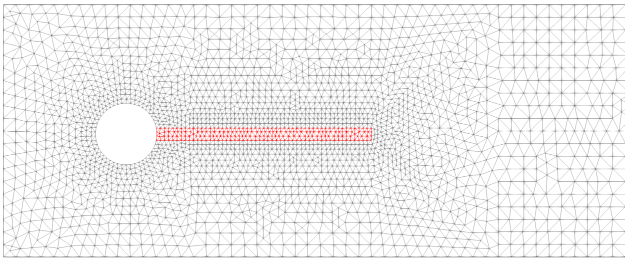
$$u_x(0, y) = 1.5\overline{u}\frac{y(H - y)}{(H/2)^2} \tag{4.13}$$

where the mean inlet velocity is $\overline{u}H$. For the following experiments, the velocity scale is selected as $\overline{u} = 2$, in accordance with the literature [192, 193]. No-slip boundary conditions are applied to the top and bottom boundaries and on the surface of the rigid cylinder. At the outlet boundary, a "do nothing" condition is applied which in the weak formulation naturally satisfies the stress-free condition $(\boldsymbol{\sigma} \cdot \boldsymbol{n} = 0)$.

An example of the unstructured triangular mesh is shown in Figure 4.13, for the coarsest case. The resolution of the mesh is scaled around the number of cells located along the vertical boundaries of the fluid domain, m. Here, the number of cells along the horizontal fluid domain is $(L/H)m$. The resolution of the mesh is finer in the region surrounding the solid, where the velocity gradients are expected to be larger. The number of cells vertically across the filament is $m_h = (3/20)m$, and along the length of the filament $(l/h)m_h$. The number of elements along the circumference of the cylinder is scaled by $(\pi/\theta - 1)m_h$, where θ is the half angle of the thickness of the filament, $\theta = \sin^{-1}(h/2r)$.



(a) Total domain.



(b) Zoomed region around the solid.

Figure 4.13: Example mesh for Turek case, mesh $m = 20$, solid region outlined in red.

For this setup, where large deflections of the solid are observed, the deterioration of the mesh quality is a concern, particularly near the points of maximum solid deflection. Remeshing of the domain could be used to ensure mesh quality is maintained throughout. For periodic responses of the solid, with high frequencies, there will be a requirement to re-mesh often (potentially more than twice per single period) at a high computational expense. To observe the direct influence of pressure on the solid response, as opposed to measuring other influencing factors, remeshing is not applied in these experiments and the mesh parameters used (refer to Table 4.5) were selected based on numerical experimentation.

Parameter		Value
Channel length	L	2.5
Channel height	H	0.41
Cylinder centre position		$(0.2, 0.2)$
Cylinder radius	r	0.05
Filament length	l	0.35
Filament thickness	h	0.02
Reference point (at $t = 0$)		$(0.6, 0.2)$

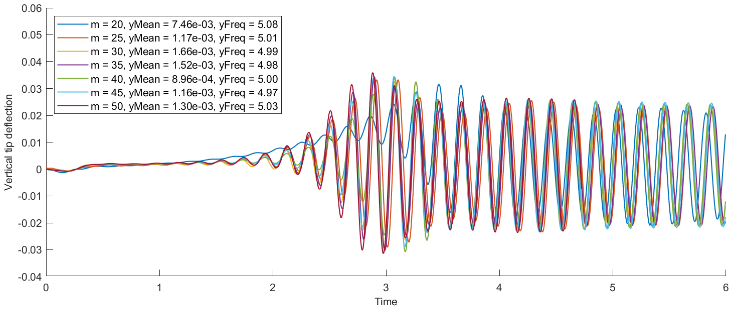
Table 4.4: Geometric parameters for Figure 4.12.

Fluid-solid parameters		
Solid density	ρ^s	1×10^3
Fluid density	ρ^f	1×10^3
Solid constant	c_1	2×10^6
Fluid viscosity	μ^f	1

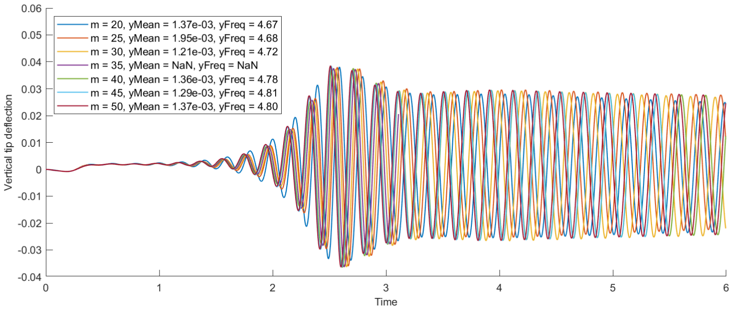
Mesh parameters		
First Lamé parameter	λ^m	10
Second Lamé parameter	μ^m	1

Table 4.5: Material properties of the fluid-solid system (left) and the elastic solid governing the mesh deformation (right), for the geometry outlined in Figure 4.12.

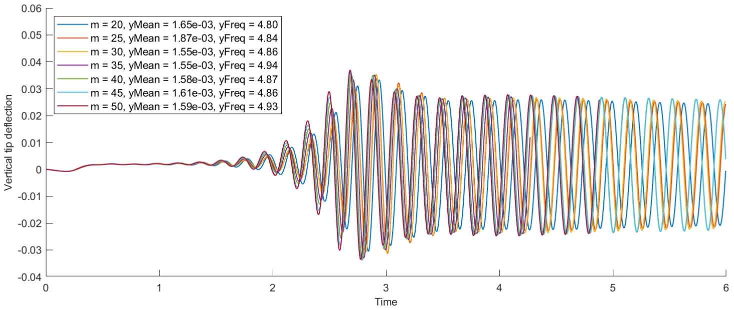
The vertical tip deflection is plotted over time for a number of mesh resolutions, for the P_2/P_1, P_2/P_0 and $P_2/(P_1 + P_0)$ spaces, shown in Figures 4.14. For the higher mesh resolution ($m = 40$ for P_2/P_1, $m = 50$ for P_2/P_0 and $m = 45$ for $P_2/(P_1 + P_0)$) the simulations were terminated prematurely and the complete $t = 6$s was not possible. In these cases, the quality of the mesh deteriorated in the fluid region near the tip of the solid filament, where the density of the mesh is high and the displacement of the mesh is large. Without the use of remeshing, this is one of the key drawbacks of the single-mesh approach of the ALE method.



(a) P_2/P_0



(b) P_2/P_1



(c) $P_2/(P_1 + P_0)$

Figure 4.14: Vertical tip displacement of the solid, $\Delta t = 0.005$. yMean is the mean amplitude of the first period of oscillation, and yFreq is the mean vertical frequency (averaged in time) of oscillation, after steady state has been reached ($t > 3.5$).

The mesh problem sizes shown in Figure 4.14 are tabulated in Table 4.6.

Mesh Level	m	Degrees of Freedom		
		Velocity	Pressure	Total
1	20	32080	7856	39936
2	25	48056	11814	59870
3	30	68676	16926	85602
4	35	97508	24090	121598
5	40	129394	32019	161413
6	45	154522	38265	192787
7	50	195390	48439	243829

(a) P_2/P_0

Mesh Level	m	Degrees of Freedom		
		Velocity	Pressure	Total
1	20	32376	4129	36505
2	25	48056	6107	54163
3	30	69220	8774	77994
4	35	97508	12332	109840
5	40	137298	17327	154625
6	45	154522	19498	174020
7	50	207742	26172	233914

(b) P_2/P_1

Mesh Level	m	Degrees of Freedom		
		Velocity	Pressure	Total
1	20	32376	12059	44435
2	25	48056	17921	65977
3	30	69220	25836	95056
4	35	97508	36422	133930
5	40	137298	51322	188620
6	45	154522	57763	212285
7	50	207742	77699	285441

(c) $P_2/(P_1 + P_0)$

Table 4.6: Degrees of freedom of Turek geometry for different mesh resolutions.

The frequencies of the filament response generally agree with the results from Turek and Hron [192]. Compared to the range of results presented in Turek et al. [193], our results are at the lower end of the predicted frequency range, as shown in Figure 4.16a. Conducting a direct comparison of the frequency of the vertical displacement of the filament for the finest mesh resolutions for all three pressure spaces, as shown in Table 4.7, we observed that it is P_2/P_0, which is in the best agreement with the literature. For amplitudes P_2/P_0 and $P_2/(P_1+P_0)$ miss predicting by around the same amount, 10.3% and 9.7% respectively, relative to the literature.

FE space	Total DoFs	a_y	f_y
P_2/P_0	243,829	1.30×10^{-3}	5.00
P_2/P_1	233,914	1.37×10^{-3}	4.78
$P_2/(P_1+P_0)$	285,441	1.59×10^{-3}	4.92
Turek et al. [193]	304,128	1.47×10^{-3}	5.46

Table 4.7: Comparison of amplitudes and frequencies of the vertical tip deflection for all three pressure spaces of the finest mesh resolution, $m = 50$, with "Method 3" as outlined in Turek et al. [193].

From Table 4.7, "Method 3" uses an ALE approach with Taylor-Hood finite elements, and a temporal resolution of $\Delta t = 2.5 \times 10^{-4}$. Turek et al. [193] solve using a discrete Newton method that solves each subsystem using a Krylov multigrid approach. Using the UMFPACK direct solver for the three pressure spaces, a memory limitation is reached at $\sim$ 2.8e5 degree of freedom, less than in the cases reported in the literature. It is noticeable that the resolution of the result generated using the direct solver has a lower resolution compared to the literature, using fewer degrees of freedom, even when comparing the same FE space (P_2/P_1).

This discrepancy in the results is likely to be primarily due to the different solid models used. In the original paper by Turek and Hron [192] a Saint Venant-Kirchhoff model for a compressible hyperelastic solid is applied, compared to the incompressible neo-Hookean model in the results shown here. The simplicity of the Saint Venant-Kirchhoff extension of Hooke's law into a nonlinear domain has been criticised as giving a poor representation for nonlinear structural analysis [117].

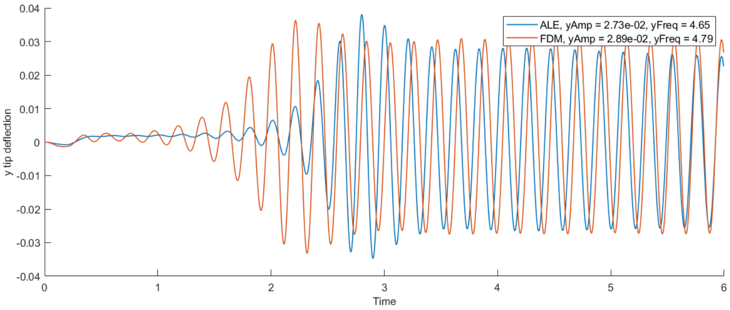
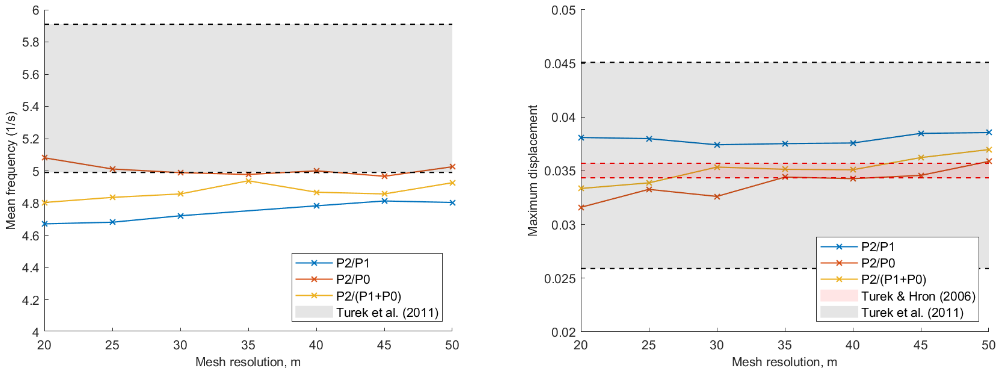


Figure 4.15: Comparison of the vertical displacements for the ALE (outlined in this thesis) and a fictitious domain method (FDM) (as outlined in Wang et al. [200]), both using P_2/P_1 FE space and $m = 20$.

Figure 4.15 shows a direct comparison between the ALE method, as detailed in this thesis, with a two-mesh fictitious domain method (FDM), using the same geometric setup and material properties. The FDM used to generate these results is described in Wang et al. [200], however, in this case without any mesh adaptivity. From the initial undeformed state, the FDM amplitude grows at a much higher rate, reaching a maximum deflection at $t = 2.2$s compared to $t = 2.8$s with the ALE method. When a steady-state oscillatory response is reached, both methods reach a close agreement in amplitude and frequency. Unlike with the ALE approach, the main source of error with FDM is the interpolation of the solid contributions onto the background fluid field.

The pressure space P_2/P_1 cannot provide a mean amplitude and frequency for the mesh resolution $m = 35$, as indicated by the 'NaN' in the legend of Figure 4.14b and the lack of data points in Figure 4.16a. In this case, the mesh quality deteriorates early with the simulation terminating prematurely, specifically for this resolution and pressure space, otherwise discrete remeshing would be required to continue.



(a) Mean frequency of the vertical of the tip deflection.

(b) Maximum vertical tip displacement.

Figure 4.16: Comparison of steady state deflection parameters using the different pressure spaces with results from the literature given in two papers by Turek and Hron [192] (in red) and Turek et al. [193] (in grey).

Since the ALE formulation, as outlined in Chapter 4, which generated these results is not energy-conserving, the amplitude of the filament tip will decay between each cycle. However, the mean amplitude and frequency do reach a steady state and are therefore comparable to the benchmarking data in the literature.

To make a comparison with previous studies, the effect of amplitude decay must be omitted, the maximum amplitude is used to make a more fair comparison to the literature, shown in Figure 4.16b. In general, all pressure spaces agree well with the literature, near the upper and lower bounds of both the papers reported by Turek and Hron [192] and Turek et al. [193] coincide with the results reported in the literature.

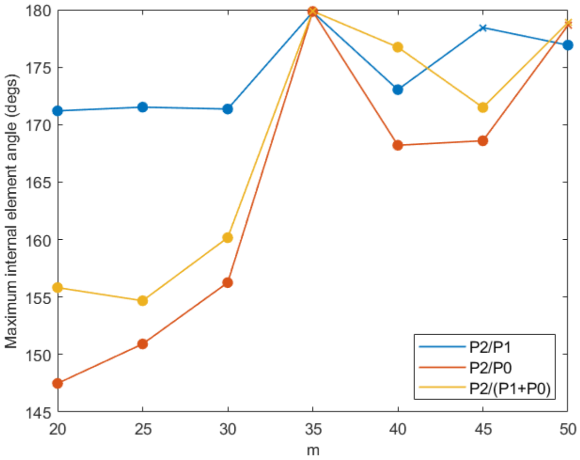


Figure 4.17: Maximum internal element angle for each resolution run; where the marker symbol × signifies that simulation terminated prematurely and ● the simulation run was completed.

In Figure 4.17 it is shown that all meshes for $m = 35$ exhibit a total deterioration of mesh quality, where an element interior angle approaches 180°. Apart from P_2/P_0 the simulations fail when the angle approaches this limit. Other cases also become problematic when an interior angle approaches 180°; P_2/P_1 for $m = 45$ and both P_2/P_0 and $P_2/(P_1 + P_0)$ for $m = 50$. Similarly, when the mesh resolution increases, this can lead to increased shearing of smaller elements in critical regions, such as in the fluid mesh around the tip of the solid. Overall, it appears that the pressure spaces which include the discontinuity tend to have improved mesh quality when measured by the maximum internal element angle.

For mesh conforming approaches, such as ALE, the quality of the mesh will affect the convergence rate, stability, and accuracy of the solution. This can be particularly important for transient, oscillatory solid deflections. An ideal mesh would consist of regular elements constructed from equilateral elements (triangles or tetrahedrons). Highly distorted meshes produce flat or "sliver" elements. To monitor the quality of the mesh, the maximum interior angle of each element is calculated. For a periodic deformation, the mesh quality varies over each cycle, with the poorest quality observed at the maximum absolute deflection.

Furthermore, the cumulative effect of repetitive cyclic deformation of the mesh leads to deterioration of the overall quality of the mesh over time.

Several approaches can be employed to rectify the deterioration of mesh quality, which are briefly discussed below.

1. Adaptive time stepping can be used to maintain the quality of the mesh to prevent mesh tangling, where the maximal discretisation length of a given element, h, is reduced.

2. Local mesh repair, for example, could consist of performing an edge swapping operation of two adjacent sliver elements, therefore, reducing the maximum angle of the given elements [76].

3. Problem-specific mesh constraints: specific points in the mesh can be constrained given *a priori* information about the expected deformation of the solid. This would allow the mesh to deform with the solid geometry in a more consistent way. This is done in a similar sense to Ruschak [168] when free surface flows are modelled.

4. Global or local remeshing: This can be computationally expensive and leads to a loss of accuracy. Therefore, it should be avoided where necessary [130].

The application of these approaches contravenes the general methodology of ALE and is outside the scope of this project. The purpose of this thesis is to understand the impact of the choice of pressure space on the accuracy and efficiency of the solution. Since we have been able to apply the solver successfully across many time steps, the decision was made not to implement discrete remeshing in these two-dimensional tests.

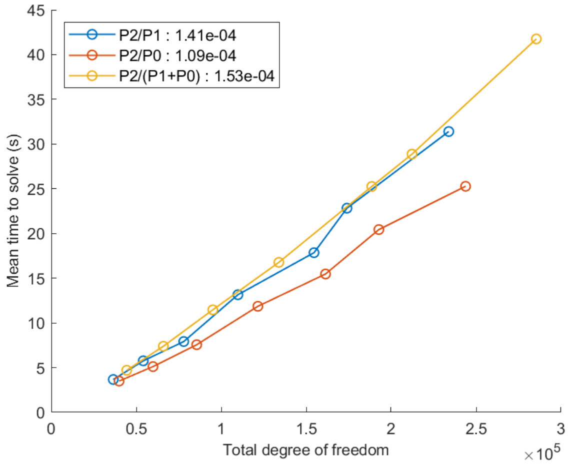


Figure 4.18: Mean solve time (per time step) over entire runs versus total number of degrees of freedom, the slope of each series is given for each FE pairing in the legend. All linear systems are solved using the UMFPACK sparse direct solver [48].

As a measure of efficiency, Figure 4.18 illustrates the time to solve the linearised system of equations at each time step, averaged over the entire simulation for each mesh resolution (represented by the total number of degrees of freedom). The gradient of these trends is a measure of how the solution time scales with the size of the problem, given this FSI case. Similar performance is observed for both P_2/P_1 and $P_2/(P_1+P_0)$ spaces, yet P_2/P_0 marginally outperforms the other spaces.

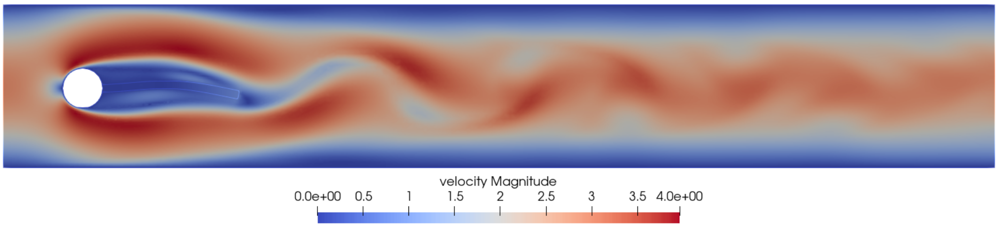
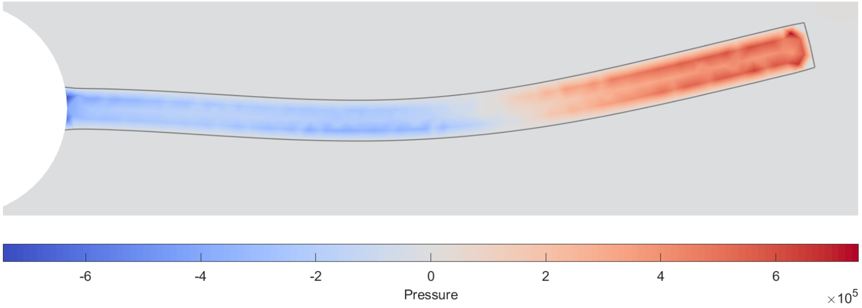
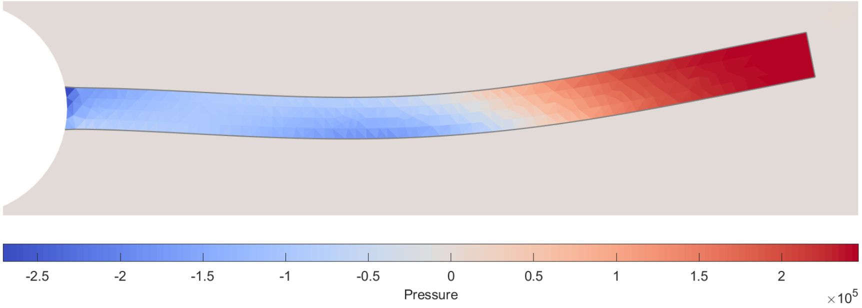


Figure 4.19: Example velocity magnitude contour plot, P_2/P_1, $m = 30$, $t = 3.50$.

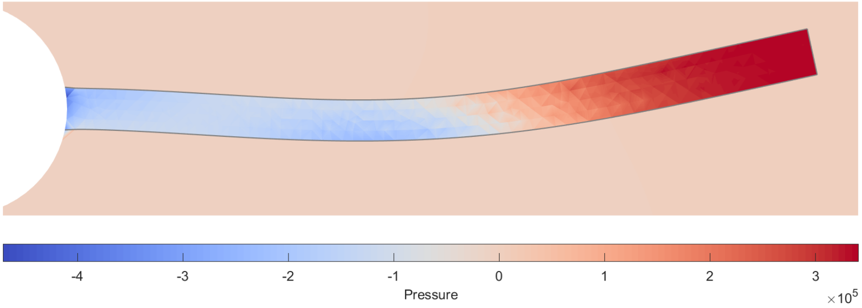
Figure 4.19 shows a snapshot of the visualisation of the velocity magnitude of the resultant flow. The fluid accelerates around the fixed solid cylinder before interacting with the deformable filament behind and generating the vortex shedding bound by the horizontal boundaries of the channel.



(a) P_2/P_1, $m = 30$, $t = 4.17$.



(b) P_2/P_0, $m = 30$, $t = 4.10$.



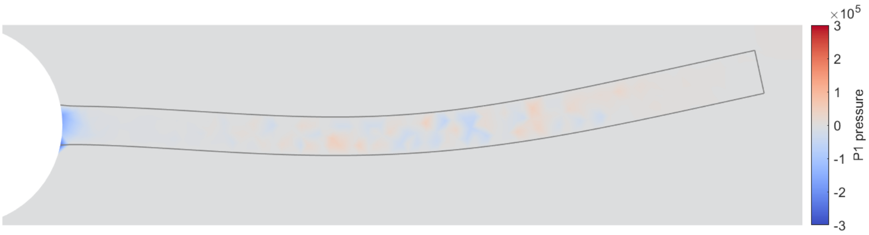
(c) $P_2/(P_1 + P_0)$, $m = 30$, $t = 4.12$.

Figure 4.20: Pressure contour plots around the solid filament, outlined in grey. The region show is bound by $x = [0.22, 0.62]$ and $y = [0.15, 0.25]$.

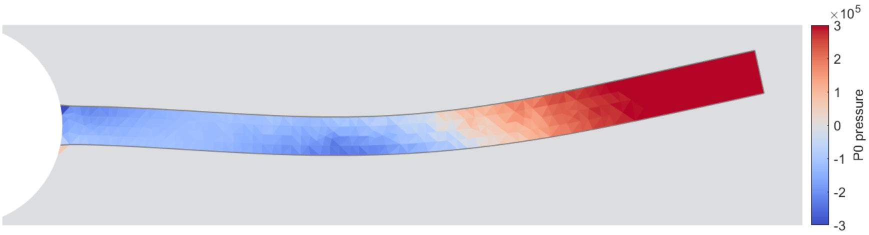
Alternatively, pressure contour plots can be generated in MATLAB, as is done in Figures 4.20, which focus on the filament region in different snapshots in time. Specifically, they show the deflection at the first maximum amplitude after $t > 4$, which occurs at slightly different times due to phase delay between pairs of FE spaces.

Across the three FE spaces, the pressure within the solid is large relative to the fluidic pressure. The pressure range in the solid for the P_2/P_1 pressure space can clearly be seen to smear across the elements near the boundary of the fluid structure. Both the P_2/P_0 and $P_2/(P_1 + P_0)$ pressure plots are similar, in terms of the distribution and magnitude of the pressure, but without this smearing. It is possible that the linear continuous component of $(P_1 + P_0)$ only has a notable contribution in regions of the solid where the cells are too large to resolve high pressure gradients using only discontinuous pressures alone, which is observed mainly in the region of the filament that is attached to the rigid cylinder.

As in the previous case, the pressure contributions are decomposed into their components in Figure 4.21. The P_1 pressure only has a small negative contribution at the base of the solid, where it is attached behind the solid cylinder. Similarly, in the mid-length region of the solid, there is a small contribution from P_1, again in a region where the pressure gradients are the greatest. Most of the pressure contribution arises from the discontinuous, P_0, component.



(a) P_1 pressure.



(b) P_0 pressure.

Figure 4.21: Pressure field plots decomposed into both the piecewise linear continuous and piecewise constant discontinuous contributions. For the same region, time step and mesh resolution as shown in Figure 4.20c.

4.4 Concluding remarks

The use of discontinuous pressure demonstrates greater accuracy in capturing the pressure across the interface, using fewer degrees of freedom. This near-interface pressure has a large impact on the deflection of the solid, thus, there is a requirement for it to be efficiently resolved.

Of the discontinuous pressure approximations, P_2/P_0 overall demonstrates better performance than $P_2/(P_1 + P_0)$. This may be in part because the enriched discontinuous pressure space includes additional degrees of freedom associated with the linear pressure contribution. Although this would suggest that the pressure is resolved to a higher order, for the problem presented in this study, it has not proven to be essential to resolve an accurate linear pressure field. In the scenario where the pressure has a high spatial gradient and is continuous, one could envisage that the high-order pressure space would be of greater importance.

Chapter 5

3D finite element spaces

The previous chapter examines the use of a discontinuous pressure finite element space to capture fluid-structure interaction in two dimensions. Given the observed benefits, the chapter aims to extend the approach to three dimensions.

One of the main differences between the two- and three-dimensional cases is the increase in the size of the problem for a like-for-like mesh resolution. Another key difference is that the use of the P_2 velocity space with the discontinuous pressure P_0 space is observed to be unstable for general three-dimensional meshes. Applying some of the previous finite element (FE) spaces, such as P_2/P_0 and $P_2/(P_1 + P_0)$, directly in three-dimensional problems leads to unphysical results. This is because, unlike with the two-dimensional cases with these pairs of FE spaces, in three dimensions there are not enough velocity degrees of freedom relative to the number of pressure degrees of freedom in each element.

When extending the analysis of the three FE spaces examined in the previous chapter to three-dimensional space, it is widely proven in the existing literature that the simplest second-order Taylor-Hood element (P_2/P_1) is Ladyzhenskaya-Babuka-Brezzi (LBB) stable [87].

To extend the discontinuous pressure space, P_0, to three dimensions and remain LBB stable, a higher order velocity approximation must be considered. Zhang [211] show that P_3/P_0 is stable on tetrahedral grids, by using the macro-element theory of Stenberg [179]. For P_2/P_0 in three dimensions, one would need a mid-face node to control the normal flux between neighbouring elements

[26]. Professor David Silvester, a Chair in Numerical Analysis at the University of Manchester and a well-published author in the field of finite elements for incompressible fluid dynamics, was contacted for advise on the use of the discontinuous pressure spaces in three dimensions. Professor Silvester states that "the issue that compromises the discontinuous pressure mixed approximation on tetrahedral meshes [is] the absence of a midface velocity node" [175]. This can be stabilised either through a mid-face bubble function or a higher-order velocity space, the latter of which naturally has a mid-face node.

The enriched Taylor-Hood element, $P_2/(P_1 + P_0)$, is found to be LBB stable in two dimensions for a Stokes problem, under the condition that each element has at least one internal node [186]. In three dimensions, Boffi et al. [28] are able to theoretically prove (Theorem 3.1) that an enriched Taylor-Hood element, $P_{m+1}/(P_m + P_0)$ is LBB stable, given $m \geq 2$. From the macro-element theory, the stability is provided by the shared velocity degree of freedom across the face of neighbouring elements, when using P_3 velocity finite element space. Therefore, it is assumed that $P_3/(P_1 + P_0)$ would therefore also be LBB stable in three dimensions, as the pressure space has fewer degrees of freedom compared to $P_3/(P_2 + P_0)$. This proof by Boffi et al. [28] was performed only for a Stokes flow; this provides a strong indication that this finite element pair will be found to be experimentally stable for the ALE FSI formulation.

Another constraint on the stability of the enriched Taylor-Hood element in three dimensions, as shown by Boffi et al. [28], is that any element has at least one internal vertex. When generating the mesh, it must be avoided that an element has all vertices located on the boundary of the domain, for example, at the right-angled corner of a regular domain.

In the book by Gresho et al. [87], they specifically state that $P_2/(P_1 + P_0)$ is indeed LBB stable for three-dimensional tetrahedrons. Although the proof from Boffi et al. [28] does not inherently disprove that $P_2/(P_1 + P_0)$ is stable, Gresho et al. [87] do not explicitly provide evidence to support their claim.

It is further worth mentioning that another difference in three dimensions, the formulation of the hyperelastic solid is also different from two dimensions because of the Cayley-Hamilton theorem, generating additional terms in the solid. We refer back to Section 3.6.2 for the full implementation.

Three stable finite element pairs, P_2/P_1, P_3/P_0 and $P_3/(P_1 + P_0)$, are introduced and tested in two three-dimensional problems. The first case considers a three-dimensional version of the lid shearing case examined in Section 4.3.1, while the second numerical experiment considers a deformable wall in an open-channel flow. The numerical experiments aim to test the performance of the three finite element spaces, with regards to the efficiency required to accurately capture the solid deformation. This chapter will only make use of sparse direct solvers for linear algebra, which restricts the size of the meshes used. The following chapter will address this constraint through the introduction of preconditioned iterative solvers.

5.1 Finite element spaces

This section introduces the chosen finite element spaces in three dimensions, showing the setup of the degrees of freedom on the tetrahedral simplex.

5.1.1 P2/P1

For three dimensions the degree of freedom layout of this element pair is naturally extended from the triangle in section 4.2.1, to a tetrahedron in Figure 5.1. With a total of 14 degrees of freedom on this element.

The properties of the P_2/P_1 element in three dimensions are retained from two dimensions. The Taylor-Hood element remains a stable choice for the discretisation of three-dimensional finite elements, fulfilling the LBB stability condition [87]. Stenberg [180] performs stability analysis on three-dimensional tetrahedral elements for incompressible flows.

5.1.2 P3 space

To provide stability with discontinuous pressure elements, the order of the velocity space can be increased relative to the order of the pressure space, using a cubic velocity approximation.

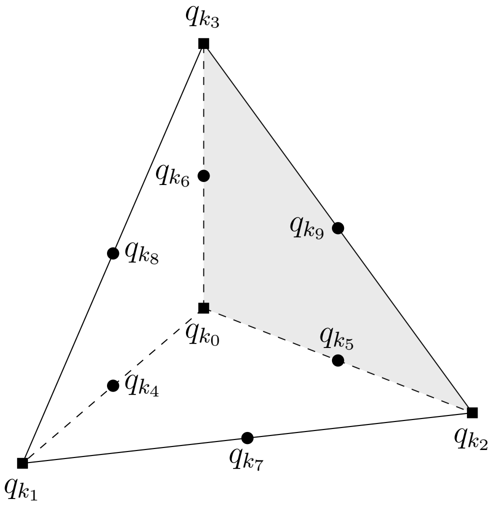


Figure 5.1: P_2/P_1 degrees of freedom on a 3D tetrahedron; ● velocity and ■ velocity and continuous pressure.

As before, the finite element space for the piecewise P_3 continuous velocity space is defined as follows,

$$V_h^3 = \{\boldsymbol{v} : \boldsymbol{v} \in H^1(\Omega),\ \forall k \in \mathcal{T}_h : \boldsymbol{v}_k \in P_3(k)\}, \tag{5.1}$$

where P_3 is the a set of polynomials of degree ≤ 3. As shown in Figure 5.2 and Figure 5.3 the extra degrees of freedom of velocity are located on the edges and at the centre of each face of the simplex tetrahedron.

P3/P0

The structure of this element is given in Figure 5.2, noting the four velocity degrees of freedom on each edge of the simplex, and a single velocity degree of freedom at the centre of each face. It is these degrees of freedom located in the centre of the faces that provide stability for the discontinuous pressure, and thus the reason why P_2/P_0 is not stable in three dimensions. This gives a total of 21 degrees of freedom for this element type.

This higher order velocity FE space benefits from being able to capture the finer velocity detail, where for slender solid geometries undergoing shearing deformation, high mesh resolutions may not be possible, combined with the discontinuous pressure space's capability to conserve mass at the element level [28].

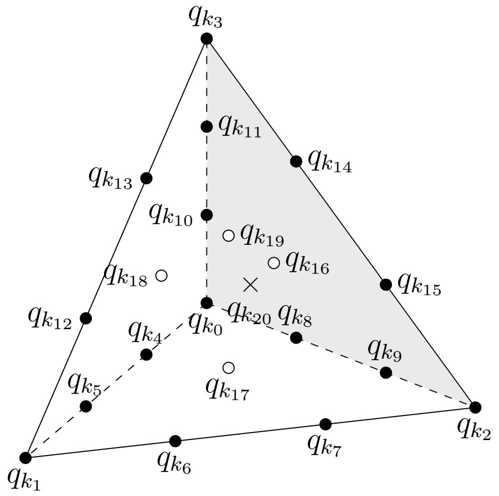


Figure 5.2: P_3/P_0 degrees of freedom on a 3D tetrahedron; ● velocity on the edges, ○ velocity on the faces, and × discontinuous pressure (located at the barycenter).

This is the lowest-order element pair which both has a velocity degree of freedom located on the centre of a face and discontinuous pressure space, without the use of bubble functions or other stabilisation approaches.

P3/(P1+P0)

Similarly for just the P_0 pressure space, to examine the enriched pressure space in three dimensions a richer velocity space must be considered compared to its two-dimensional equivalence. The structure of this type of element is shown in Figure 5.3, with a continuous pressure degree of freedom at each vertex, for a total of 25 degrees of freedom on the simplex.

The combination of the high order velocity and richer pressure space increases the number of degrees of freedom per element. This leads to more degrees of freedom on a like-for-like mesh than for the other FE pairs discussed in this chapter. However, it is hoped that discontinuous pressures will allow a lower mesh resolution than P_2/P_1 elements for equivalent accuracy.

As with the two-dimensional $P_2/(P_1 + P_0)$ pair, this finite element has two hydrostatic modes, which both must be constrained using a mean pressure constraint.

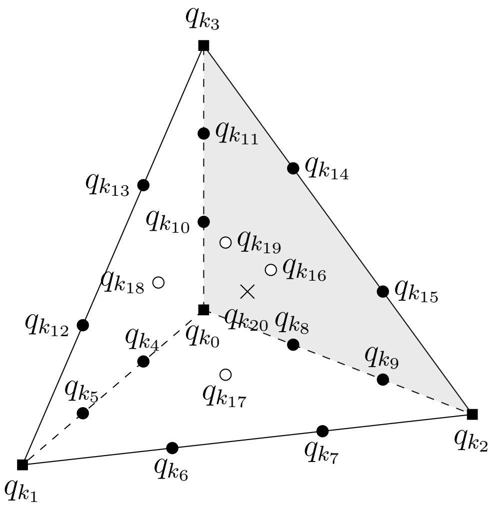


Figure 5.3: $P_3/(P_1+P_0)$ degrees of freedom on a 3D tetrahedron; ● velocity on edges, ○ velocity on faces, ■ velocity and continuous pressure and × discontinuous pressure (located at the barycenter).

5.2 Numerical experiments

Two three-dimensional numerical examples are considered to demonstrate the performance of the three finite-element spaces; P_2/P_1, P_3/P_0 and $P_3/(P_1+P_0)$, in terms of capturing an accurate deflection solution for the solid.

5.2.1 3D Test 1: Filament in a shearing cavity

This test case is a three-dimensional extension of 2D Test 1: Filament in a cavity flow, in Section 4.3.1. The two-dimensional filament is extruded circumferentially to become a three-dimensional cylinder, and a bi-quadratic lid velocity is used to drive the flow (see Figure 5.4, which shows the profile that is chosen to ensure conformity with the zero velocity boundary on the side walls). The walls at the sides and on the bottom of the domain are prescribed with impermeable, no-slip boundaries. The solid filament (shaded grey) is attached to the bottom face, located centrally in the channel, at $(L/2, H/2, 0)$. The tip of the solid cylinder (the centre of its top face) is the point of reference for measuring deflection, located at $(L/2, H/2, l)$.

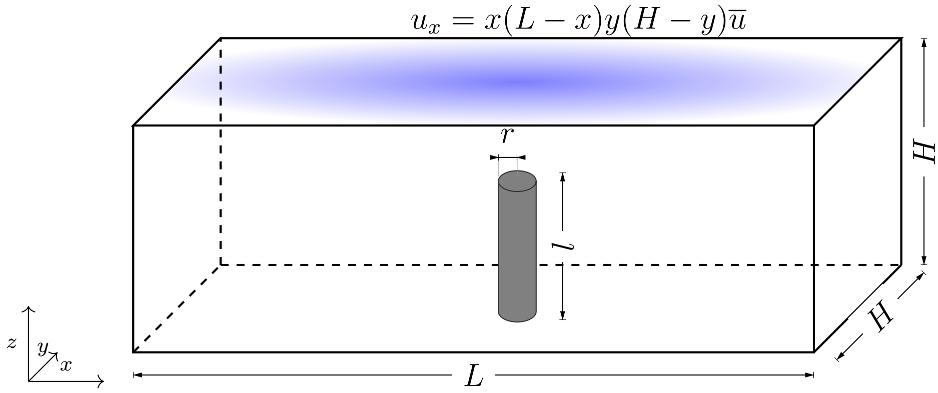


Figure 5.4: A three-dimensional extension to model presented in Figure 4.4, deformable solid highlighted in the region of grey and the lid driven by a bi-quadratic velocity.

The geometric dimensions of the test case presented in Figure 5.4 are given in Table 5.1.

Parameter		Value
Channel length	L	2.0
Channel height	H	1.0
Cylinder radius	r	0.15
Cylinder height	l	0.8

Table 5.1: Geometric parameters for Figure 5.4.

The lid velocity is defined by a biquadratic x-component shear, $u_x(x,y) = x(L-x)y(H-y)\overline{u}$, where the velocity is scaled by $\overline{u} = L^2H^2$ to ensure unity in the centre of the face. To avoid an initial discontinuity at startup, a gradual increase in the shear velocity is prescribed through a time varying profile,

$$u_x(x,y,t) = \begin{cases} \frac{u_x(x,y)}{2}(1-\cos(2\pi t)), & t < 0.5 \\ u_x(x,y), & t \geq 0.5. \end{cases} \tag{5.2}$$

The model is run, using the parameters in Table 5.2, to a steady state deflection at $t_f = 1$.

Fluid-solid parameters		
Solid density	ρ^s	10
Fluid density	ρ^f	10
Solid constant	c_1	1×10^5
Fluid viscosity	μ^f	10

Mesh parameters		
First Lamé parameter	λ^m	1000
Second Lamé parameter	μ^m	100

Table 5.2: Material properties of the fluid-solid system (left) and the elastic solid governing the mesh deformation (right), for the test case outlined in Figure 5.4.

The computational mesh for the three-dimensional cases are unstructured tetrahedral meshes, generated using TetGen [174] which is called within FreeFEM [93]. The resolution of the mesh is scaled around the number of cell increments, m, along the length H in Figure 5.4. To ensure the quality of the meshes produced in TetGen, the radius-edge ratio is prescribed with an upper limit of 1.2. The fluid-solid interface is explicitly captured within the domain. An example of the computational mesh generated is shown in Figure 5.5, and the mesh statistics for a number of the meshes used in each test are given in Table 5.3. Examining the same mesh resolution, $m = 10$, the number of pressure degrees of freedom for the lower-order space, P_3/P_0, are ~ 4.6 times greater relative to the linear pressure space of P_2/P_1.

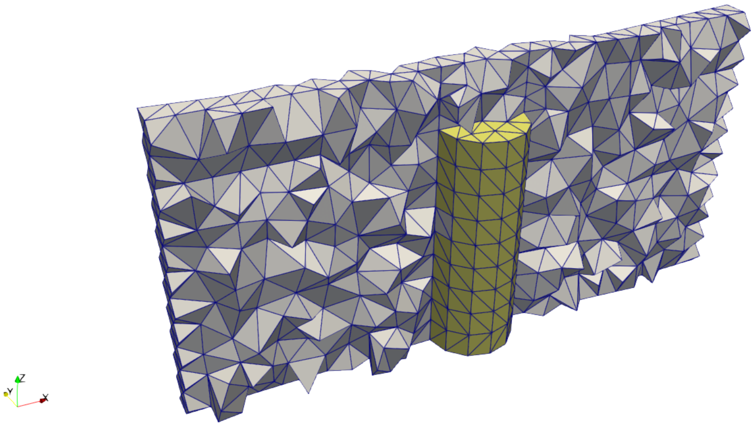


Figure 5.5: Example of the computational mesh, slice through the entire domain at $y = 0.5$ and the outline of the solid region (in yellow).

Mesh Level	m	Degrees of Freedom			% difference
		Velocity	Pressure	Total	
1	8	37662	1824	39486	-
2	10	63591	2984	66575	-2.7329
3	12	108276	5022	113298	-6.9667
4	14	164667	7502	172169	-4.2887
5	16	233280	10488	243768	-2.6960
6	18	333738	14791	348529	-1.5370

(a) P_2/P_1

Mesh Level	m	Degrees of Freedom			% difference
		Velocity	Pressure	Total	
1	8	120000	7809	127809	-
2	10	204963	13681	218644	-4.1147
3	12	350214	23527	373741	-2.0738

(b) P_3/P_0

Mesh Level	m	Degrees of Freedom			% difference
		Velocity	Pressure	Total	
1	8	120000	9633	129633	-
2	10	204963	16665	221628	-5.1401
3	12	350214	28549	378763	-2.7785

(c) $P_3/(P_1 + P_0)$

Table 5.3: Degrees of freedom, with percentage difference in horizontal tip deflection between subsequent mesh refinements.

Figure 5.6 shows the horizontal deflection of the solid filament in the lid-driven flow, for each mesh resolution. As the mesh resolution increases, successive improvement in steady-state deflection reaches an asymptotic mesh-converged solution. This is further demonstrated in Figure 5.7. In fact, from Figure 5.7, the $P_3/(P_1 + P_0)$ performs similarly to P_2/P_1, in terms of deflection with degrees of freedom. Although P_3/P_0 converges at a similar rate, it tends to a different steady-state deflection, compared to P_2/P_1 and $P_3/(P_1 + P_0)$. The lower order

P_0 pressure space alone may not be sufficiently accurate, compared to the linear contributions of P_2/P_1 and $P_3/(P_1 + P_0)$ FE spaces.

Although the upper boundary imposes a velocity in the positive x-direction, the solid deflects in the negative x-direction, as shown in Figure 5.6. The fluid accelerates in the boundary layer near the lid, its general motion recirculates within the enclosed domain; as demonstrated in Figure 5.10. Consequently, most of the fluid impinging on the solid filament is moving in the negative x-direction, and the solid deflects in the corresponding direction.

When using the higher-order cubic velocity space, the number of degrees of freedom grows significantly faster than with quadratic velocities in two dimensions for each mesh resolution. Combined with the three pressure spaces examined, there is a computational limit to using a sparse direct solve, for any given fixed hardware[1]. In our tests, the memory capacity of the hardware limits the problem size to approximately 100,000 degrees of freedom when solving three-dimensional problems with sparse direct methods. Hence, for this chapter, we only consider problems up to this size (see Chapter 6 for a discussion of preconditioned sparse iterative solves).

---

[1] In this chapter numerical experimental were performed locally on a desktop PC, in serial, equipped with an Intel Xeon Processor E3-1240 v5, with four double threaded core (total eight threads), and a base clock speed of 3.50 GHz. A total of 15.4 GB RAM memory was available.

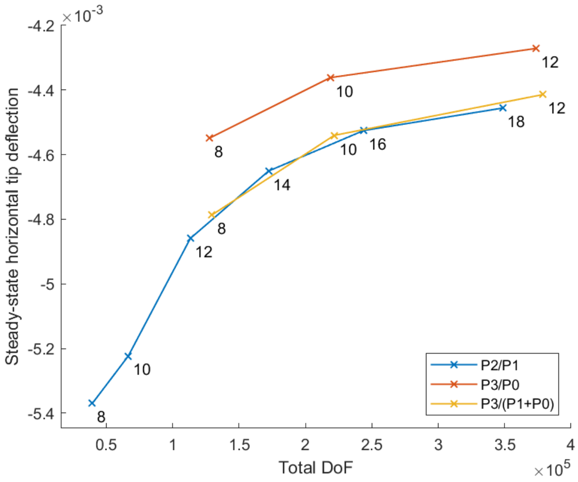


Figure 5.7: Steady state deflection with number of total degree of freedom for each pressure space. The number next to each node is the value of m.

Figure 5.8 shows increase of memory usage when solving the FSI linear system with the number of degrees of freedom, and Figure 5.9 is the mean solve time. Memory is measured as the resident set size before and after the solve routine is called, obtained within the FreeFEM script. From both of these figures, we observe that the P_2/P_1 space is less efficient for a given number of degrees of freedom, it requires more memory and solving time relative to the P_3/P_0 and $P_3/(P_1 + P_0)$ spaces.

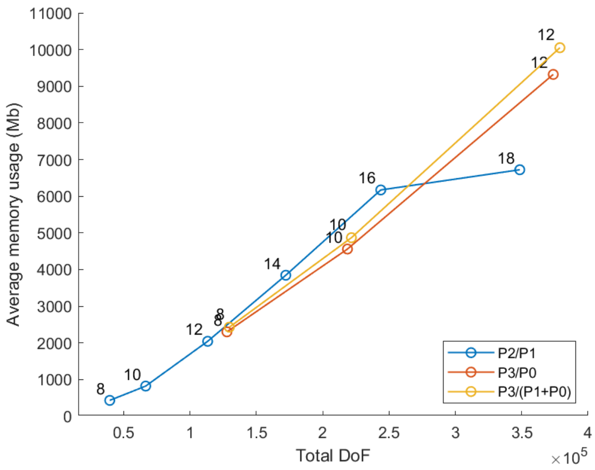


Figure 5.8: Mean memory used during the solve of each linear system for each pressure space.

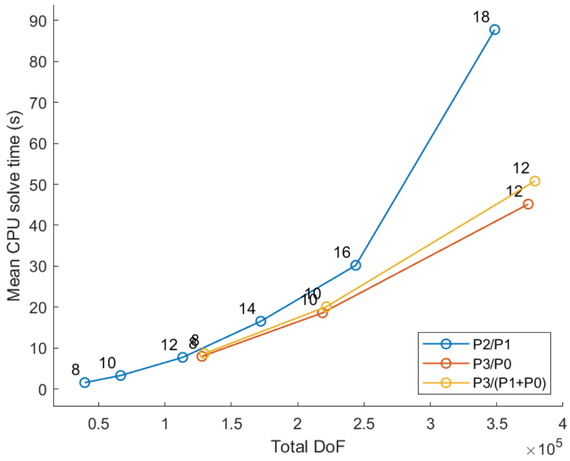


Figure 5.9: Mean solve time for each linear system for each pressure space.

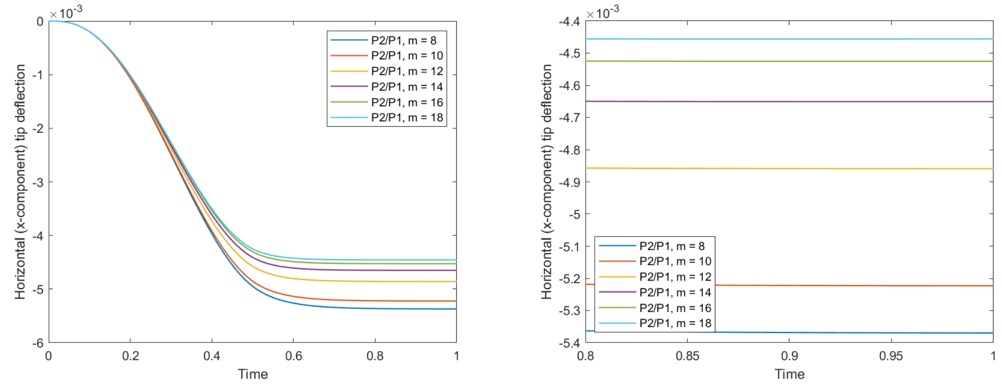
As was observed in the two-dimensional case, the mesh for $m = 18$ (for P_2/P_1) could be so dense that the quality of the elements degrades so that the system becomes poorly conditioned, causing a sudden increase of solve time (shown in Figure 5.9) but a reduction in memory usage (shown in 5.8). Ignoring the anomalous data points at $m = 18$ for the P_2/P_1 FE space in Figure 5.8, the rate of growth of memory usage initially looks linear, growing at a rate of $0.0286N$, where N is the number of degrees of freedom of the system, which is related to the size of the problem. On closer inspection, the trend appears to follow a polynomial growth rate more closely, close to a quadratic rate, precisely at a rate of $N^{1.38}$. As a measure of the closeness of the fit, the coefficient of determination calculated for the linear fit is 0.989, compared to 0.998 for the superior polynomial fit.

The mean solve time does not grow linearly, however, super-linearly, and again by ignoring the $m = 18$ case (for P_2/P_1) the trends are calculated. The P_2/P_1 FE space grows at a rate of $N^{1.80}$. For P_3/P_0, which is an improvement in the solve time rate, at $N^{1.71}$. However, for $P_3/(P_1 + P_0)$ solve time grows rate deteriorates, measured at $N^{1.84}$. Although both FE spaces with discontinuous pressures only use three data points, it is difficult to determine how accurate these trends are without more data.

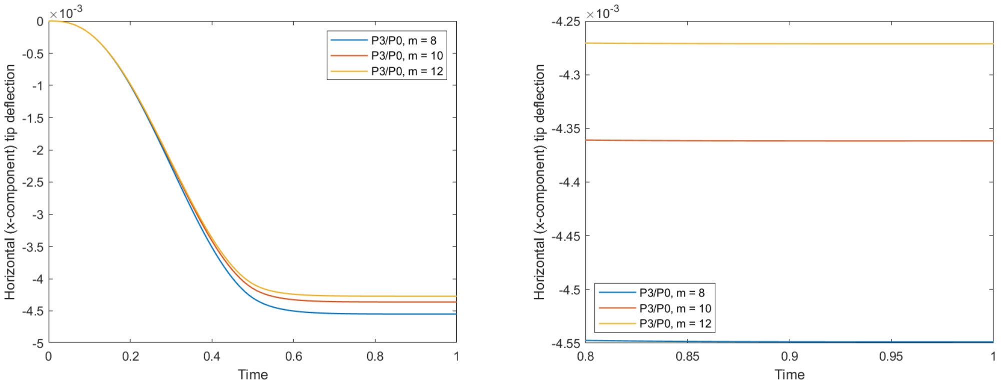
Neither the P_3/P_0 and $P_3/(P_1 + P_0)$ FE pairing could provide enough data to examine this trend, when using direct solvers.

Typically, inverting a dense matrix using Gauss elimination, as in LU decomposition, would cost $O(N^3)$ operations. Here, the discrete system is sparse, and its inverse (which will be denser) is not explicitly stored. To solve using LU decomposition for a sparse matrix, one would expect much better performance, provided that the fill-in is controlled. PETSc calls the multifrontal sparse solver package MUMPS when performing an LU direct solve, this aims to order the rows in the elimination process so as to minimise fill-in.

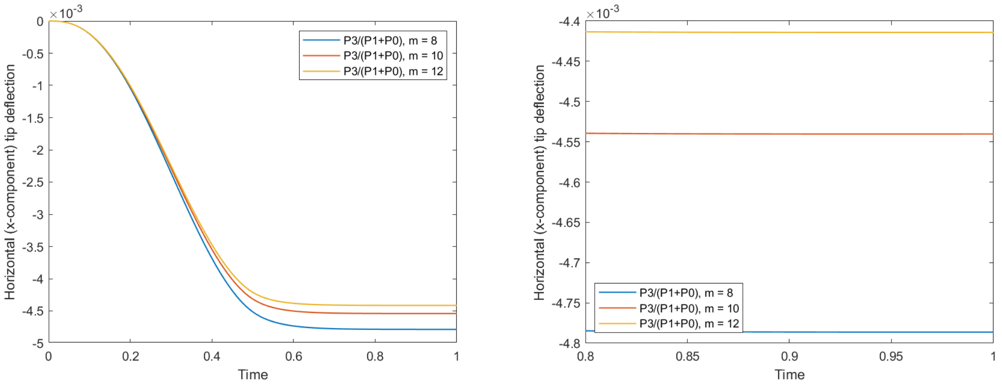
Similarly, examining the growth in memory usage with problem size, a fit of $N^{1.24}$ is determined for the P_2/P_1 finite element space.



(a) P_2/P_1



(b) P_3/P_0



(c) $P_3/(P_1 + P_0)$

Figure 5.6: Horizontal tip deflection in the three dimensional shear driven cavity for a given mesh resolution. Figures on the left show the complete time series, and figures on the right show the final $t \geq 0.8$.

Figure 5.10 shows the velocity streamlines as the fluid recirculated in the cavity, deflecting around the deforming solid. Unlike the two-dimensional case, where the fluid generated two circulation regions either side of the solid filament, in three dimensions the fluid is able to bypass the solid leading to the much smaller deflection.

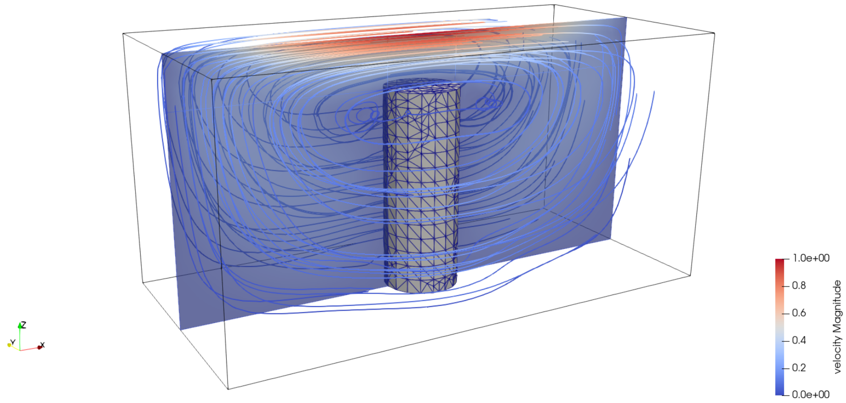
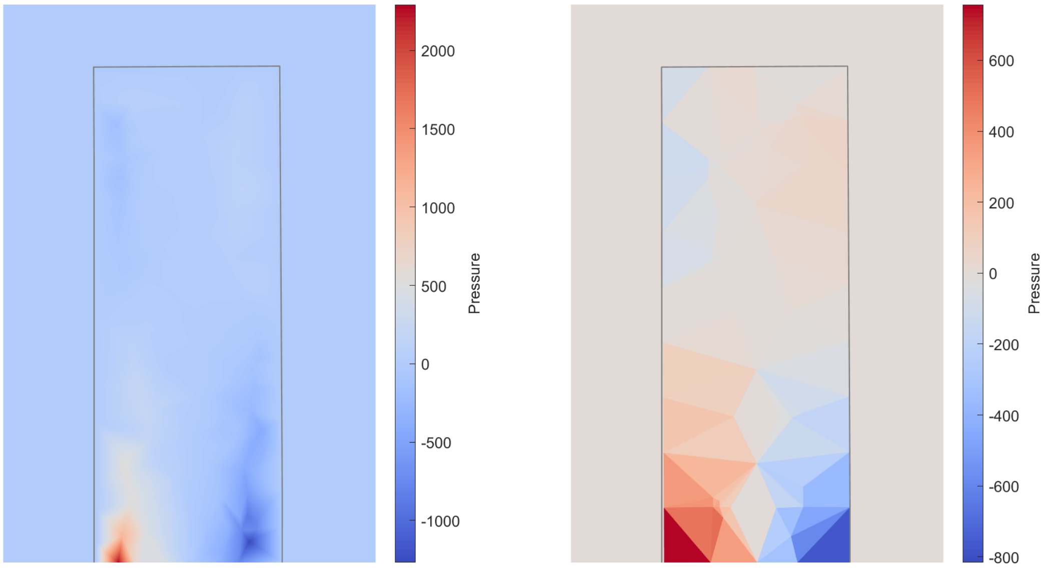


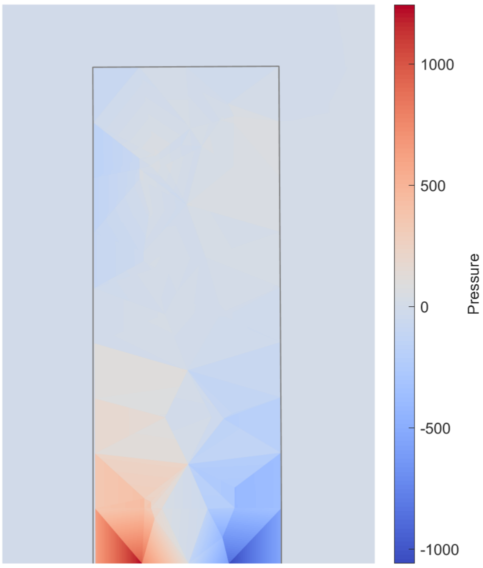
Figure 5.10: Velocity streamline around the deformation filament, using $m = 18$, P_2/P_1, at time $t = 1.0$.

As in the two-dimensional cases, ParaView cannot plot the discontinuous linear pressure functions. Therefore to visualise $P_3/(P_1 + P_0)$ the solution is interpolated on a two-dimensional plane, through the point $p_0 = (1, 0.5, 0.5)$ with a normal $p_n = (0, 1, 0)$, using MATLAB. A search is performed that identifies where the edges of the tetrahedron element intersect the display plane, constructing a two-dimensional mesh of a mixture of triangles and quadrilaterals. Linear interpolation of the P_1 pressure solution is used to project the solution onto the plane, whereas the P_0 solution which is constant on the three-dimensional elements is also the same for their two-dimensional projections.



(a) P_2/P_1, $m = 18$, $t = 1.0$.

(b) P_3/P_0, $m = 12$, $t = 1.0$.



(c) $P_3/(P_1 + P_0)$, $m = 12$, $t = 1.0$.

Figure 5.11: Pressure contour plots around the solid filament of a cut view at $y = 0.5$, showing the regions $x = [0.7, 1.3]$ and $z = [0, 0.9]$. The solid region is outlined in grey.

Across all pressure spaces, in Figures 5.11, the background fluid pressure remains at a magnitude of the order of one. It is observed that the pressure jump at the fluid-structure interface is clearly captured in Figures 5.11b and 5.11c, which is particularly large close to where the solid attaches to the boundary. Whereas in Figure 5.11a the pressure is slightly smeared at the offset to the interface, within the solid filament. In the case of the enriched pressure space, Figure 5.11c, the linear varying pressure is clearly observable in the larger elements at the base of the filament. A maximum pressure is measured in the filament adjacent to the fluid-structure interface, however, located where the solid is attached to the boundary.

5.2.2 3D Test 2: Leaflet in channel flow

The second 3D test case considers a deformable wall obstruction in a cross-channel flow. The solid is again fixed to the bottom face, however the wall does not extend the total width of the channel ($w < H$), see Figure 5.12. The bottom face has a no-slip boundary condition applied ($\boldsymbol{u} = 0$), with the front ($u_y = 0$), back ($u_y = 0$) and top faces ($u_z = 0$) with slip boundary conditions. At the inlet and outlet faces a normal velocity profiles prescribed,

$$u_x(z) = \frac{3}{2}\frac{z}{H}\left(2 - \frac{z}{H}\right)\overline{u}, \tag{5.3}$$

and time-varied as,

$$u_x(z,t) = \begin{cases} \frac{u_x(z)}{2}(1 - \cos(2\pi t)), & t < 0.5 \\ u_x(z), & t \geq 0.5, \end{cases} \tag{5.4}$$

where the velocity is scaled to unity $\overline{u} = 1$.

The geometric dimensions of the test case presented in Figure 5.12 are given in Table 5.4. The dimensions of the fluid region remain the same as the problem presented in section 5.2.1, with a different solid geometry and fluid forcing approach. The regularity of the solid in relation to the fluid domain makes the meshing more uniform, and less susceptible to sudden changes in element sizes or skewness.

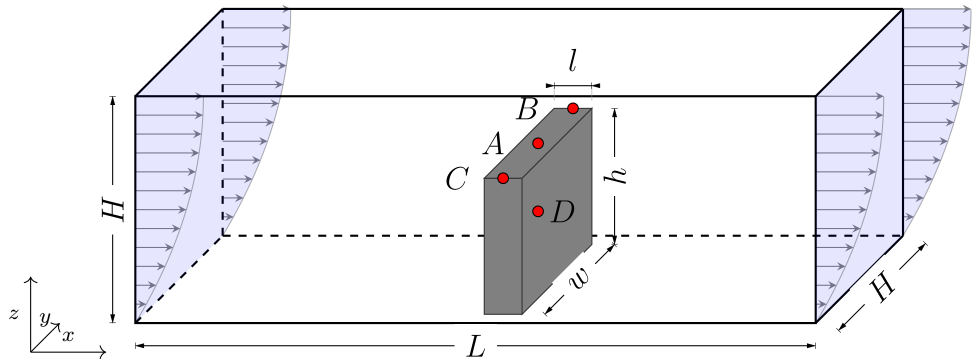


Figure 5.12: This setup considers a similar fluid domain to that in Figure 5.4, periodic boundaries are applied at the end and the symmetric boundary condition on the top. Further, the cylindrical filament is replaced with an orthogonal cuboid. Red points indicate the location of deflection monitors.

Parameter		Value
Channel length	L	2.0
Channel height	H	1.0
Wall length	l	0.2
Wall width	w	0.8
Wall height	h	0.8

Table 5.4: Geometric parameters for Figure 5.12.

An example of a computational mesh for this three-dimensional test case, mesh level 1 in Table 5.5, is shown in Figure 5.13. The three-dimensional meshes for this case are generated with TetGen [174], within PETSc, and visualised using ParaView.

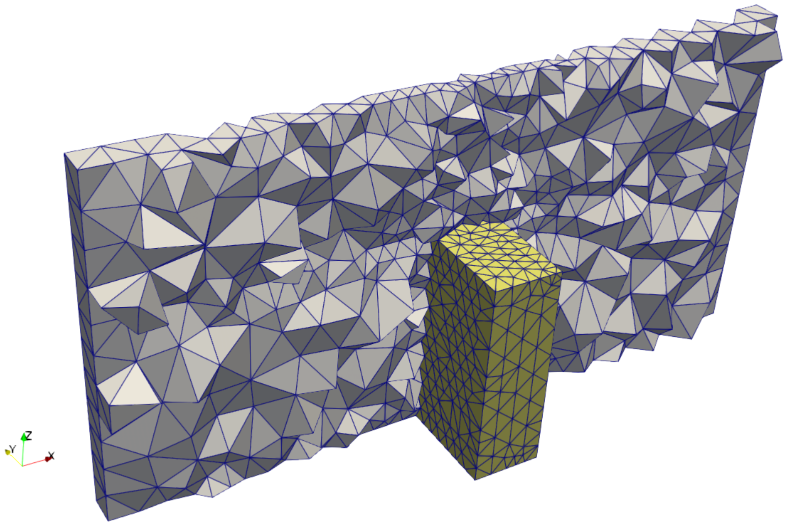
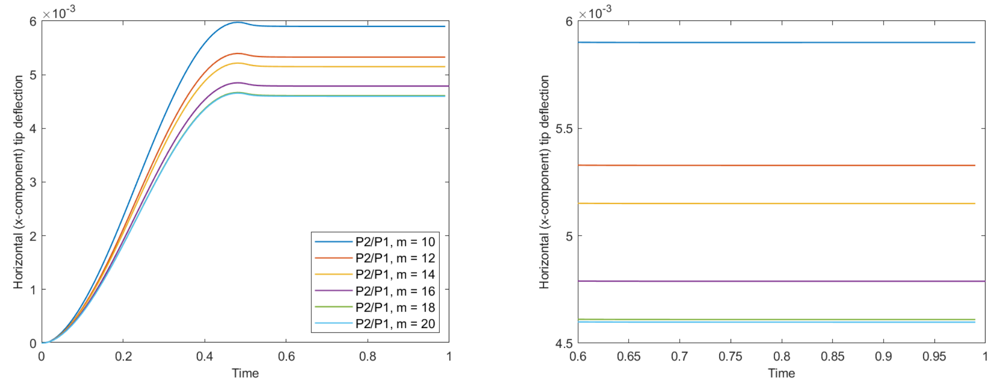


Figure 5.13: Example of the computational mesh, slice through the entire domain at $y = 0.5$ and the outline of the solid region (in yellow).

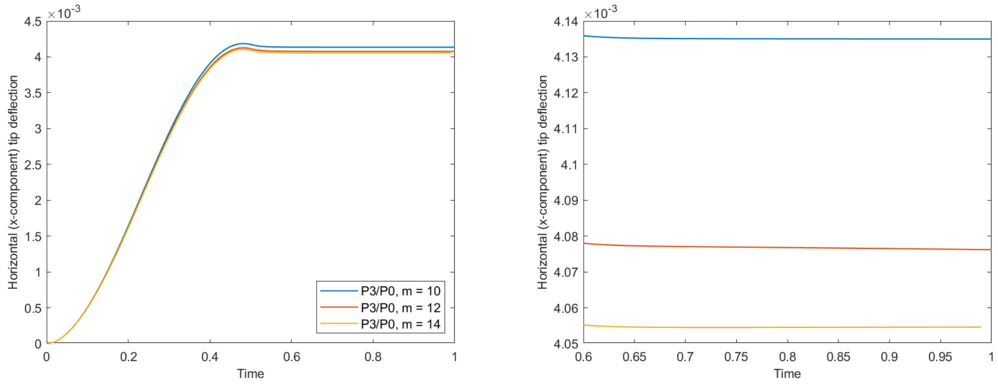
Four points are distributed within the solid to record the different modes of the deflection, with the following coordinates.

Point A	$(L/2, H/2, h)$
Point B	$(L/2, (H+w)/2, h)$
Point C	$(L/2, (H-w)/2, h)$
Point D	$(L/2, H/2, h/2)$

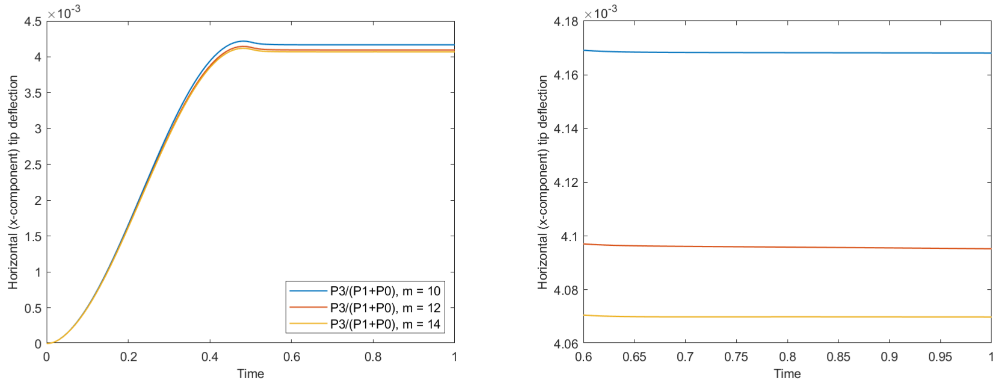
The deflection of point A is shown in Figure 5.14 for each pressure space and mesh resolution. As the velocity increases, the deflection of the solid also increase upon reaching a steady-state deflection when the velocity reaches a maximum. Note that for all mesh cases, the maximum deflection occurs slightly before reaching steady-state, due to the inertia carrying the solid deflection beyond the steady-state point. The number of degrees of freedom and the changes in steady-state solid deflection for each mesh resolution are given in Table 5.5.



(a) P_2/P_1



(b) P_3/P_0



(c) $P_3/(P_1 + P_0)$

Figure 5.14: Horizontal tip deflection in the three dimensional channel flow for a given mesh resolution. Figures on the left show the full time series, and figures on the right show a zoom in the final stages, for $t \geq 0.6$.

Mesh Level	m	Degrees of Freedom			% difference
		Velocity	Pressure	Total	
1	10	47622	2222	49844	-
2	12	78966	3590	82556	-9.7016
3	14	117645	5286	122931	-3.3335
4	16	165201	7342	172543	-7.0252
5	18	228351	10065	238416	-3.7240
6	20	308286	13477	321763	-0.2697

(a) P_2/P_1

Mesh Level	m	Degrees of Freedom			% difference
		Velocity	Pressure	Total	
1	10	153858	10329	164187	-
2	12	257406	17605	275011	-1.4218
3	14	384873	26503	411376	-0.5298

(b) P_3/P_0

Mesh Level	m	Degrees of Freedom			% difference
		Velocity	Pressure	Total	
1	10	153858	12551	166409	-
2	12	257406	21195	278601	-1.7474
3	14	384873	31789	416662	-0.6197

(c) $P_3/(P_1 + P_0)$

Table 5.5: Degrees of freedom, with percentage difference in horizontal tip deflection between subsequent mesh refinements.

Figure 5.15 show the steady-state deflection of point A, with the number of degrees of freedom. With each successive refinement of the mesh, the deflection solution converges to a mesh-independent result. The difference in solution between the two mesh resolutions of the two discontinuous pressure spaces, P_3/P_0 and $P_3/(P_1 + P_0)$, is already small. It is difficult to determine whether these are truly mesh independent without more observations.

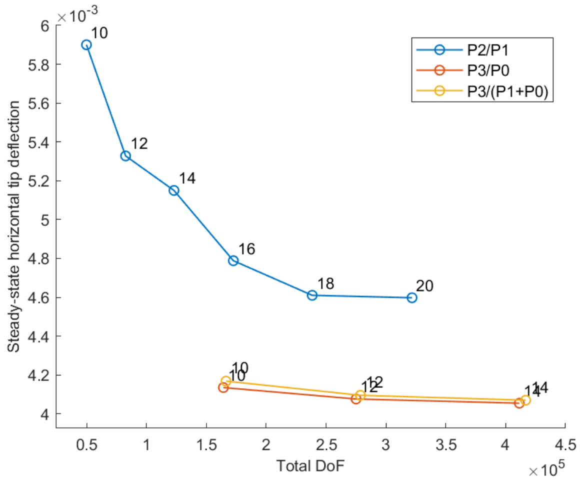


Figure 5.15: Steady-state deflection with number of total degrees of freedom for each pressure space.

Figure 5.16 show the relation of the mean memory requirement to solve the linear system with the total degrees of freedom using a direct solver, and Figure 5.17 show the relation of the mean time to solve. As in the previous case, the finite element spaces, which include the discontinuous pressure, are faster to run and use less memory, particularly for larger meshes. P_2/P_1 has a linear memory dependency on the number of degrees of freedom, up to the $m = 18$ case where performance begins to deteriorate. The time to solve grows close to quadratically, precisely the solution time grows at $O(N^{1.86})$.

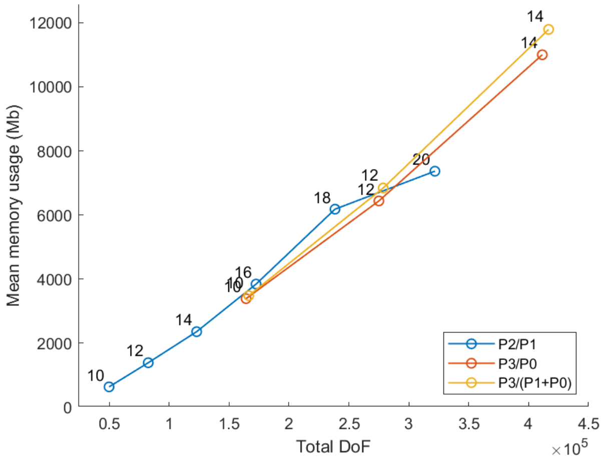


Figure 5.16: Mean memory used during the solve of each linear system for each pressure space.

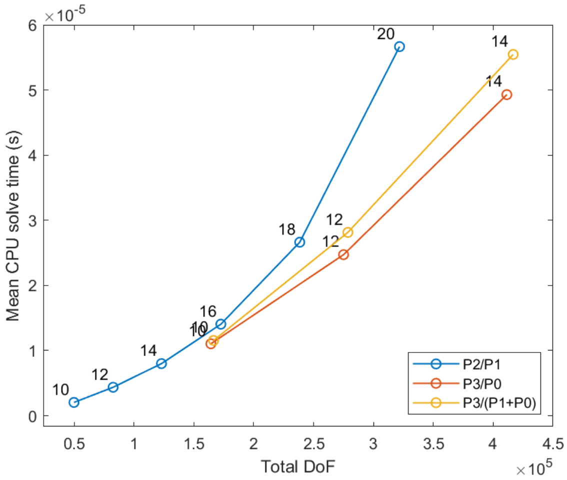


Figure 5.17: Mean solve time for each linear system for each pressure space.

The flow visualisation of the three-dimensional case with the solid wall is displayed in Figure 5.18, showing a horizontal and vertical aligned sheet of streamlines. The flow is ducted around the side and top of the solid wall, where the velocity is accelerated given the change in the streamwise cross-sectional area. The slip conditions applied to the boundaries of the fluid channel do not influence the vertical flow profile, while a small boundary layer forms around the solid.

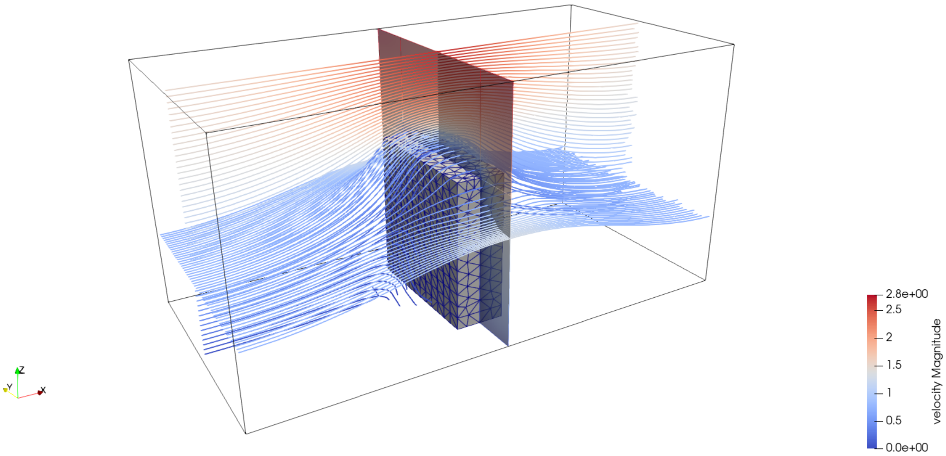
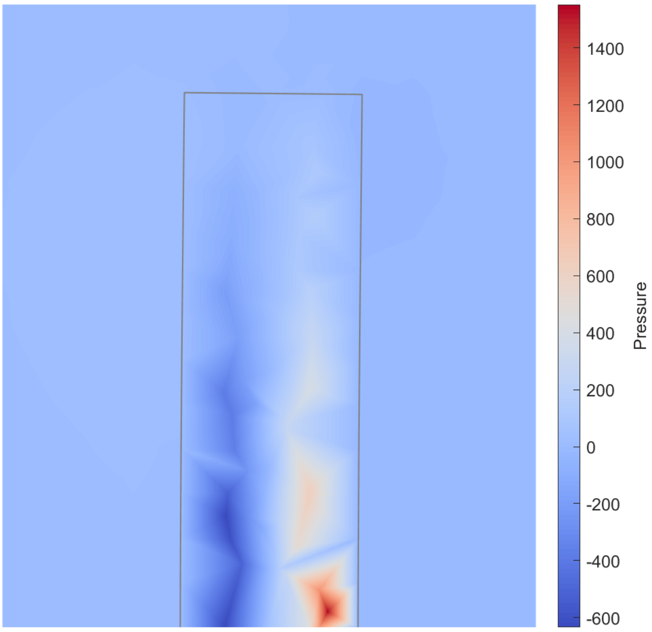
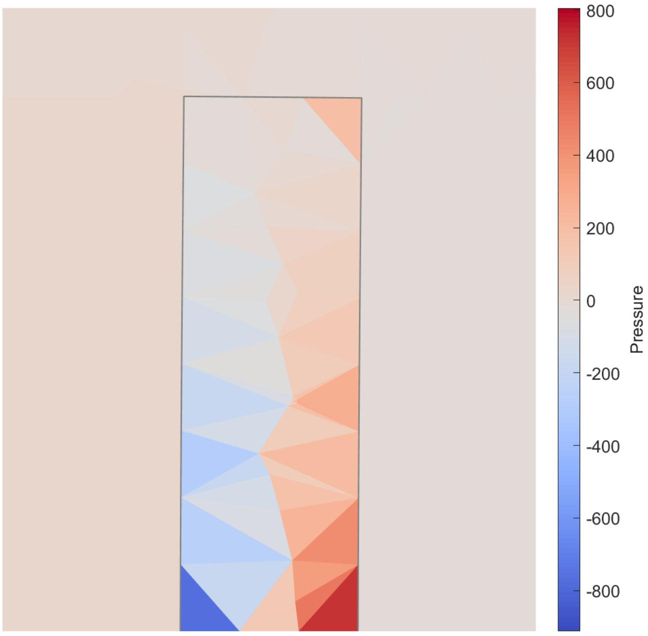


Figure 5.18: Velocity streamline around the solid wall in the channel flow, using mesh $m = 20$, P_2/P_1, at time $t = 0.99$.

Pressure contour slices taken at $y = 0.5$ are shown in Figures 5.19. They show such coarse resolution that, in some cases, only two pressure degrees of freedom are resolved across the width of the solid. Qualitatively, the pressure distribution of P_3/P_0 and $P_3/(P_1 + P_0)$ are very similar, suggesting that the most significant improvement is the discontinuous pressure component, P_0. In the region around the base of the solid, where pressure gradients are large and cells are sizeable, the linear pressure component plays a more significant role. In addition, for more complex fluid flows, the linear pressure component could have a consequential role in resolving the pressure.

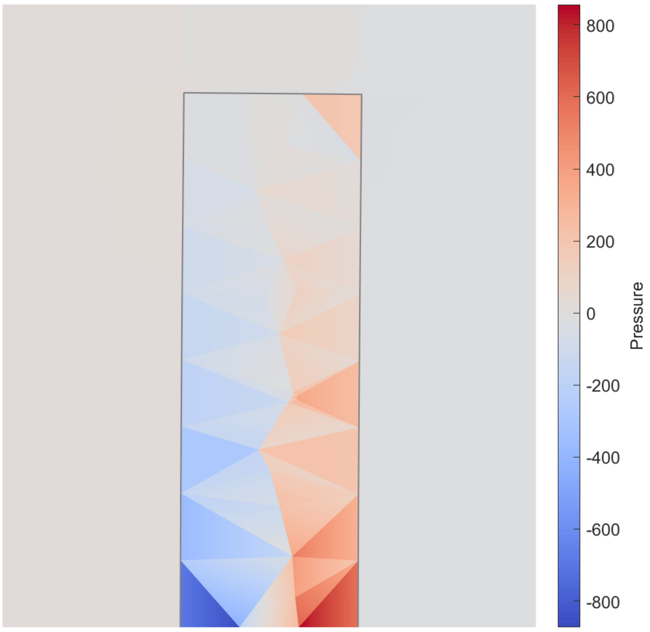


(a) P_2/P_1, $m = 20$, $t = 1$.



(b) P_3/P_0, $m = 14$, $t = 1$.

Figure 5.19: Pressure contour plots around the solid filament using a slice at $y = 0.5$, displaying the regions around the solid $x = [0.7, 1.3]$, $z = [0, 0.7]$.



(c) $P_3/(P_1+P_0)$, $m=14$, $t=1$.

Figure 5.19: Pressure contour plots around the solid filament using a slice at $y=0.5$, displaying the regions around the solid $x=[0.7,1.3]$, $z=[0,0.7]$.

5.3 Concluding remarks

For small three-dimensional problems, sparse direct solvers are applicable, which scale sufficiently well to solve moderately sized linear systems. Increasing the resolution for three-dimensional problems, particularly where FSI requires conformal fitting meshes, the number of degrees of freedom grows rapidly (for a completely regular grid and domain, the mesh size could grow as $O(m^3)$, assuming constant grid spacing in all dimensions of the domain). In this case direct solvers are not scalable, since the solution time scales super-linearly, with number of degrees of freedom. Even more significantly, however, the constraining feature for these direct solvers is the required memory, where the fill-in from factorisation of the sparse matrix can generate enormous memory requirements, which soon exceed those available on a standard desktop workstation. The fill-in from the use of sparse direct solvers is made even more significant by the use of cubic velocity spaces, as required by the fact that lower-order velocity spaces FE pairs, P_2/P_0

and $P_2/(P_1+P_0)$, are not LBB stable in three dimensions. The alternative to the sparse direct solvers are iterative methods.

A drawback of using discontinuous pressure spaces, P_0, relative to the continuous pressure spaces, P_1, is that there are typically more degrees of freedom, although they resolve a lower-order pressure solution for the same mesh resolution. Since for the P_0 pressures the degrees of freedom are associated with each cell, the number of cells outnumbers the nodes, where the nodes are shared between multiple cells, a feature that is particularly notable for three-dimensional tetrahedral meshes.

Conversely, these drawbacks are outweighed by the main benefit of using discontinuous pressures: A more accurate pressure solution at the interface of solid and fluid. This additional accuracy of the pressure solution permits a more accurate deflection of the solid, leading to solid deflection convergence at much lower mesh resolutions.

Although iterative methods offer an opportunity to solve larger problems due to the reduced memory required, they are sensitive to the structure of the linear system. This typically manifests itself in issues with convergence; either slow convergence or divergence. In addition, iterative methods will converge a solution only to the given tolerance, introducing a loss of accuracy. This motivates the demand for efficient iterative methods, where an appropriate preconditioner for the FSI application will need to be considered.

Chapter 6

Preconditioned iterative methods for FSI

Small two-dimensional FSI problems can generally be solved with sparse direct methods. For larger problems, with high mesh resolutions, particularly when refining around the interface, direct solvers are inefficient and have large memory requirements. In three dimensions, practical problems produce large linear systems and calculating solutions in reasonable times requires a more efficient solution approach.

Iterative methods offer an increased efficiency (in terms of solve time) relative to direct solvers, but their utility is governed by the number of iterations required to convergence and cost per iteration. This requires preconditioning, a strategy designed to improve the convergence rates of an iterative method. The preconditioner must be sufficiently cheap to build and apply within the iterative solve routine.

In this chapter, typical preconditioning strategies will be reviewed and then an approach is presented in the context of the ALE FSI formulation. The applied preconditioner is a block diagonal algebraic multigrid, applied to each component of the velocity, and an approximation of the Schur complement for the pressure block.

6.1 Iterative solvers

High performance iterative methods typically belong to two main classes of methods, *multilevel methods* and *Krylov subspace methods* [66]. Although a sparse direct solver can apply advanced strategies such as sparse elimination methods, which can involve a frontal or multi-frontal approach, and reordering strategies (refer back to Section 2.4.1 for further discussion), they are severely limited by the memory constraints of a given computer specification when solving. These methods are typically constrained to problems of the order of tens of thousands of degrees of freedom, and are competitive for two-dimensional problems; however, in three dimensions we required an iterative approach.

6.1.1 Multilevel methods

For multilevel methods, multigrid is the most synonymous, using a hierarchy of grids defined *a priori*, generating a smaller problem to be solved on, before interpolating (or prolongating) this solution back to the finer grid. For regular grids, this *geometric multigrid* approach is straightforward. However, for unstructured grids, which are more popular for complex deforming geometries, the hierarchy definition becomes complex.

To avoid the need for mesh properties (such as the nodal locations and the connectivity matrix) to perform the coarsening, an approach could be applied directly to (discrete sparse) algebraic system of equation generating a hierarchy of subsets of variables, without the need for prior geometric information. Thus, *algebraic multigrid* (AMG) provides a robust and efficient method to solve large classes of discrete systems, from structured and unstructured meshes, in two and three dimensions [182].

As a solver, multilevel methods (when tuned appropriately) can act optimally for discrete elliptic-type problems (Poisson problems), where the convergence rate of an iterative solver is independent of the size of the mesh [66, 182].

In addition to this, AMG has been applied successfully to a number of different non-symmetric problems (such as convection diffusion problems) and certain indefinite systems [182]. For example, as a solution approach, AMG has been applied to the Navier-Stokes equations in multiple different cases [88, 131, 203].

6.1.2 Krylov subspace methods

These classes of methods are based on the fact that, for a given sparse coefficient matrix, its product with a vector can be calculated cheaply. The Krylov subspace is defined as the following,

$$\mathcal{K}_k(A, b) := \text{span}\,(b, Ab, A^2b, ..., A^{k-1}b), \tag{6.1}$$

where $\mathcal{K}_k(A, x)$ is the linear space becoming a k-dimensional subspace of $\mathbb{R}^n$, with these vectors being linearly independent when A is non-singular.

A Krylov subspace method is based on the sparsity of the coefficient matrix A, and the product with a vector can be cheaply computed. If m is the maximum number of nonzeros in each row, and n is the number of unknowns ($A \in \mathbb{R}^{n\times n}$), then Ax can be computed with only nm floating-point operations. Thus, successive vectors, $A(Ax)$, are calculated at the same cost (nm). Although this broad class of methods aims to iteratively convergence on a solution, different approaches possess specific properties and applications. The most popular Krylov iterative methods are conjugate gradient (CG), minimal residual (MINRES), and generalised minimal (GMRES) methods.

Consider a generic large, linearised system of equations,

$$Ax = b, \tag{6.2}$$

the k-th iteration lies in the translated Krylov subspace,

$$x^0 = \text{span}\{r^0, Ar^0, A^2r^0, ..., A^{k-1}r^0\}, \tag{6.3}$$

where the initial residual is defined as $r^0 = b - Ax^0$. Thus the k-th residual vector satisfies

$$r^k \in r^0 + \text{span}\{Ar^0, A^2r^0, ..., A^kr^0\}, \tag{6.4}$$

where the Euclidean norm, $||r^k||$, is minimised over all vectors from this space.

The constraints of CG are twofold; the matrix must be both symmetric and positive definite. The CG iterate is a linear combination of vectors generated by the Lanczos algorithm, which computes the basis for the Krylov subspace $\mathcal{K}(A, r^0)$. Although CG is limited to symmetric matrices, a generalised form, the biconjugate gradient (BiCG), was developed to overcome this. The asymmetric matrix is altered to become a symmetric system $A^TAx = A^Tb$. However, the condition number of A^TA is the square of the condition number of A, therefore, this is often a much less effective method.

Similarly to CG, in that the matrix must be symmetric, the MINRES algorithm is a robust method to find a solution to an indefinite, as well as a symmetric matrix. Similarly to CG, MINRES applies the Lanczos method to orthogonal bases of the Krylov subspaces. For symmetric positive definite A, the new orthogonal vector $v^{(k+1)}$ can be calculated using a short recurrence, from previous vectors v^k and $v^{(k-1)}$.

For symmetric systems, CG and MINRES the Lanczos algorithm is used to find the orthogonal basis of the Krylov subspaces, whereas GMRES instead uses the Arnoldi method for asymmetric matrices. The GMRES method requires the construction (and storage) of orthogonal vectors that form the basis of the Krylov space, $\mathcal{K}_k(A, v^1)$,

$$\{v^1, v^2, ..., v^k\}, \tag{6.5}$$

on iteration k. To obtain a subsequent orthogonal basis, $v^{(k+1)}$, it is necessary to use all the vectors calculated previously in the computation. This can cause a practical issue in terms of memory storage. In fact, the work and storage requirements of GMRES grow like $O(kn)$ [66]. For optimal performance, the basis length should be maximised, yet this can be prohibitively expensive. To mitigate this, the inner loop of algorithm often involves a restart of the basis construction, using the final basis as the initial vector for a new loop. This is known as restarted GMRES.

In Figure 6.1, the two main branches at the top of the diagram refer to the two main methods for constructing the orthogonal basis of $\mathcal{K}_k(A, b)$. The two rows refer to the different types of conditions are established for the residual, the lower row are methods which minimise a residual norm, and the upper row using a solution of the projected matrix. FOM refers to the full orthogonalisation method and QMR is the quasi-minimal residual method.

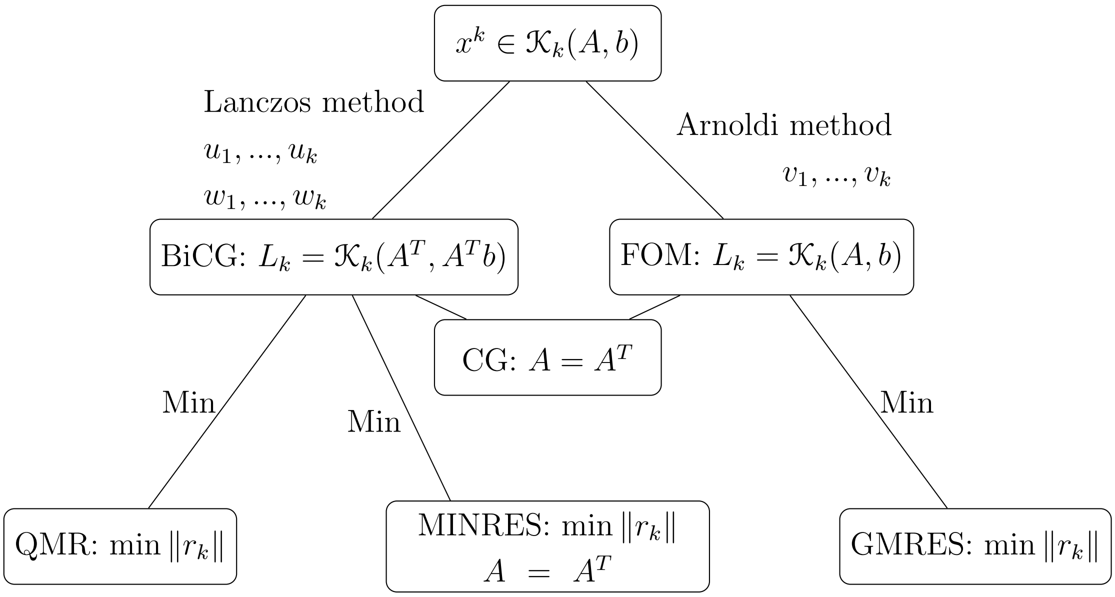


Figure 6.1: Map of Krylov subspace iterative methods. Inspired by Figure 1. from Celledoni et al. [40].

6.2 Preconditioned Krylov methods

Through the clustering of eigenvalues, the aim of preconditioners is to reduce the number of iterations to a convergence tolerance and thus reducing the overall solve time to obtain an acceptable approximation of the solution. The overall objective is to generate an approximation of the coefficient matrix A in (6.2), which requires the least amount of work to generate and apply. Realistically, it is the design of the preconditioner, M, or more precisely, the action of M^{-1} on the system that is required.

The application of left and right preconditioners is described in Subsection 2.4.3. When applying preconditioners to a symmetric solver, either CG or MINRES, it is important to maintain the symmetry of the system. Given a positive-symmetric definite preconditioner, which can be described as $M = HH^T$. Where the symmetric solver will take the form

$$H^{-1}AH^{-T}y = H^{-1}b, \tag{6.6}$$

$$y = H^T x. \tag{6.7}$$

The matrix $H^{-1}AH^{-T}$ remains symmetric and positive-definite. When preconditioning the Krylov subspace iterative method, the preconditioner is integrated into the algorithm. Specifically for a preconditioned GMRES method, the residuals are also required to be preconditioned within the Arnoldi algorithm. Also, again in the approximation of the solution step of the algorithm.

6.2.1 Incomplete factorisation preconditioning

For large, three-dimensional problems, a sparse direct factorisation solver, for example, LU factorisation, will become prohibitively expensive with increased mesh resolution as a result of the increased amount of fill-in. Zero entries of the original matrix's sparsity pattern will become nonzero as a result of the factorisation process. To mitigate this affect, incomplete factorisation approaches are considered as an approximation. The sparsity pattern of the factors is relaxed, either by retaining the sparsity of the original system or by a drop tolerance. Clearly factors which are too sparse, which will make much more affordable preconditioners, will lose resemblance to the original system and will not improve convergence rates.

These are a popular basis for out-of-the-box preconditioners since they are simple to implement and general to apply.

6.2.2 Multilevel preconditioning

Instead of using multigrid methods as a solution method, they can be applied in an approximate sense to generate a useful preconditioner.

In the same way as using multilevel methods as effective solvers, they can be effective preconditioners for other iterative methods (Krylov subspace methods). Since multigrid methods can offer precise control of the trade-off between the rate of convergence and numerical work, it makes a good tool for preconditioning. In the interest of efficiency, a multigrid preconditioner will typically involve a coarse approximation, such as a single V-cycle being sufficiently effective [119]. Another benefit of using multigrid methods as preconditioners is that it makes their performance and robustness less sensitive to the selection of certain components, such as coarse grid solvers [154].

6.2.3 Block preconditioning

For many systems of equations, the discretisation can naturally form a block structure. A physics-based block preconditioning approach is an appealing choice for flow problems. The linearised system of incompressible fluid equations, as discussed in Chapter 2, written in the form,

$$\begin{bmatrix} \boldsymbol{A} & B^T \\ B & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix}. \tag{6.8}$$

The block coefficient matrix can be factorised into a lower triangular, diagonal, and upper triangular matrix,

$$\mathcal{A} = \mathcal{L}\mathcal{D}\mathcal{U} = \begin{bmatrix} I & 0 \\ B\boldsymbol{A}^{-1} & I \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & \boldsymbol{A}^{-1}B^T \\ 0 & I \end{bmatrix}, \tag{6.9}$$

where the Schur complement takes the form $S = -B\boldsymbol{A}^{-1}B^T$. Different preconditioners can be proposed on the basis of this structure and the properties of the iterative method that is being preconditioned. The preconditioning problem that must be solved at each iteration is decomposed into smaller block systems, but it is essential that these sub-block systems can be solved efficiently themselves. If one considers the system of equations in (6.8) formed from an incompressible fluid problem, it has the properties of being symmetric and indefinite. To be able to apply an appropriate solver, such as MINRES, a preconditioner $\boldsymbol{M}$, must maintain these properties so that the matrix $\boldsymbol{M}^{-1}\mathcal{A}$ remains symmetric.

Therefore an appropriate block preconditioner could just consider the diagonal factor $\mathcal{D}$ of the block system in (6.9),

$$\boldsymbol{M}^{sym} = \begin{bmatrix} \boldsymbol{A} & 0 \\ 0 & S \end{bmatrix}, \tag{6.10}$$

to maintain the symmetry of the original matrix. In the case of applying this as a preconditioner, only the action of the inverse (or approximate inverse) of the diagonal blocks is required.

A more complex preconditioner, which would typically further accelerate rates of convergence, using the product of both diagonal and upper triangular factors, $\mathcal{D}\mathcal{U}$,

$$\boldsymbol{M}^{asym} = \begin{bmatrix} \boldsymbol{A} & B^T \\ 0 & -S \end{bmatrix}. \tag{6.11}$$

Although this asymmetric form of a block preconditioner will provide improved convergence properties [66], it is only applicable with the iterative method for asymmetric systems, such as GMRES. In a purely fluid case, when the Schur complement is found exactly in (6.11), it has been proven that an iterative solver would converge within two iterations and in the case of the exact symmetric form (6.10) within three iterations [147].

In practice it is not practical to find the exact inverse of each sub-block, and one must make further approximation to design an efficient preconditioning approach, either an approximation to the blocks or to the solution of the sub-block system. This can include diagonal scaling, incomplete factorisations, and multigrid techniques. The choice of method reflects the required accuracy of the approximation balanced with computational constraints. These computational constraints include the memory and time required to calculate and apply these preconditioners.

Consider the symmetric preconditioner (6.10), an efficient approximation to solve the sub-block system is required,

$$\begin{aligned} S\boldsymbol{z}_2 &= \boldsymbol{r}_2 \\ \boldsymbol{A}\boldsymbol{z}_1 &= \boldsymbol{r}_1. \end{aligned} \tag{6.12}$$

The matrix $\boldsymbol{A}$ is the discretisation of a second-order differential operator, and hence multigrid techniques are optimal. Hence the solution of this system can be replaced with an approximate solve using a multigrid cycle to approximate it efficiently.

The matrix S is expensive to form exactly. For a Stokes problem, it is known that the Schur complement is spectrally equivalent to the finite element sparse pressure mass matrix, $\boldsymbol{M}_p$ [66], therefore, as an approximation to this block can be replaced by the solution of $\boldsymbol{M}_p\boldsymbol{z}_2 = \boldsymbol{r}_2$. With this mass matrix, this system now becomes simple to solve since it possesses good properties, which are symmetric positive definiteness. A Cholesky factorisation would provide an exact inverse of the pressure mass matrix. The pressure mass matrix can be approximated using a fixed number of iterations of an iterative solver, such as conjugate gradient (CG), of a crude preconditioner such as just the diagonal of the pressure mass matrix. The eigenvalues are bounded as the the mesh is refined provided the matrix is diagonally scaled. The degree of approximation can be controlled through the number of CG iterations.

Considering the non-symmetric preconditioner of (6.11), applying the preconditioner retains the same approximations as previously discussed, with the addition of an extra matrix-vector multiplications,

$$\begin{aligned} S\boldsymbol{z}_2 &= \boldsymbol{r}_2 \\ \boldsymbol{A}\boldsymbol{z}_1 &= \boldsymbol{r}_1 - B^T\boldsymbol{z}_2. \end{aligned} \tag{6.13}$$

Approximation to S and $\boldsymbol{A}$ are performed as before with the symmetric preconditioner.

Fundamentally, there are two requirements for the block preconditioner, an efficient approximation to the inverse of the velocity block and the inverse of the Schur complement. Although the inexact Schur complement may again involve the inverse of the velocity block, this is often a looser approximation than for the $\boldsymbol{A}$ entry in the matrix. Further, the approximation to the Schur complement should retain spectral properties of the exact Schur complement.

6.3 Preconditioned Krylov methods for FSI

The FSI approach adopted in this work, as described in Section 3.8, leads to a discrete system with a structure similar to the Stokes problem, when using the Characteristic-Galerkin method to approximate the total derivative [66]. However, in this case the velocity block contains contributions from the solid model in addition to a contributions from the fluid. As a result, it cannot be assumed that a multigrid approach will work or that the corresponding Schur complement matrix is spectrally equivalent to the mass matrix, as has been implemented with pure-fluid problems. However, we can take inspiration from the work on the Stokes problems and use that to guide the initial development of a suitable block preconditioner. We will test this preconditioner using a single velocity-pressure pair, P_2/P_1, and we will consider the enriched pressure FE space later in this chapter.

6.3.1 Taylor-Hood element

Considering the full system with Taylor-Hood elements (P_2/P_1), including the velocity unknowns, pressure unknowns and the Lagrange multiplier related to the mean pressure constraint (refer back to (3.54)),

$$\begin{bmatrix} \boldsymbol{A} & B^T & 0 \\ B & 0 & \boldsymbol{v}_p^T \\ 0 & \boldsymbol{v}_p & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \\ \lambda \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \\ 0 \end{pmatrix}, \tag{6.14}$$

which can be decomposed into upper, lower and diagonal components through Gaussian elimination,

$$\mathcal{A} = \mathcal{L}\mathcal{D}\mathcal{U} = \begin{bmatrix} I & 0 & 0 \\ B\boldsymbol{A}^{-1} & I & 0 \\ 0 & \boldsymbol{v}_p S_p^{-1} & I \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & 0 & 0 \\ 0 & S_p & 0 \\ 0 & 0 & S_\lambda \end{bmatrix} \begin{bmatrix} I & \boldsymbol{A}^{-1}B^T & 0 \\ 0 & I & S_p^{-1}\boldsymbol{v}_p^T \\ 0 & 0 & I \end{bmatrix} \tag{6.15}$$

where $S_p = -B\boldsymbol{A}^{-1}B^T$ is the conventional Schur complement associated with the pressure block. An additional Lambda-Schur complement takes the form $S_\lambda = -\boldsymbol{v}_p S_p^{-1}\boldsymbol{v}_p^T$, associated with the mean pressure constraint. If the eigenvalue associated with this extra row and column lies outside the main cluster of eigenvalues associated with the rest of the system, it will only require a single extra iteration of the Krylov solver. The vector row $\boldsymbol{v}_p$ (and the column vector $\boldsymbol{v}_p^T$) is the discrete mean pressure constraint, as discussed in Section 3.6.2.

For the sake of simplicity, we consider the block diagonal form as an candidate structure of the preconditioner,

$$M = \begin{bmatrix} \boldsymbol{A} & 0 & 0 \\ 0 & S_p & 0 \\ 0 & 0 & S_\lambda \end{bmatrix}. \tag{6.16}$$

The same decomposition outlined in this Section can be applied to other unenriched pressure finite element spaces, such as P_3/P_0.

6.3.2 Approximation to the velocity block

The first required component of the block preconditioner is cheap a yet effective approximation to the inverse of the velocity block. The discrete velocity block includes contributions from the mass and stiffness terms, from equation (3.90) in Section 3.8, throughout the domain, with additional contribution from the discrete solid in regions of the domain occupied by the solid.

Multigrid methods proved to be fast iterative solvers for general elliptic problems. In the case of determining a preconditioner, the requirement of finding an accurate solution is no longer relevant. AMG can provide a suitable solution approach that can provide acceptable accelerated convergence for problems that may have “Poisson-like” characteristics [64]. A fixed number of multigrid cycles can be an effective preconditioner even for problems where it is not possible to generate a convergent multigrid solution [66]. If a cost-effective strategy involves a limited number of AMG cycles that are able to tighten the clustering of eigenvalues, this will improve the convergence rate of a Krylov iterative solver.

In relation to a linear elastic problem, Mihajlović and Mijalković [141] effectively applies a component-wise AMG preconditioner in conjunction with a Krylov iterative method in both two and three dimensions, demonstrating a constant iteration count with the size of the problem. They deduce from Korn’s inequality that the block diagonal preconditioned discrete elasticity operator has a solution time that scales linearly with the discretisation size [141]. This concept of using AMG to approximate the diagonal blocks of displacement components of a linear elastic solid model is applied as an optimal preconditioner for an ALE FSI system [146].

When comparing the formulation of a linear elastic model (3.55) (for the movement of the mesh), with the hyperelastic model (3.72), in this case an incompressible neo-Hookean model, we notice some similarity. In the weak form, the deviatoric part of the stress tensor is identical, $\mathrm{D}\boldsymbol{u} : \mathrm{D}\boldsymbol{v}$, the double-dot product with the test function. The double-dot product of two matrices can be calculated as the trace of the inner product. Therefore, this term contributes directly to the diagonal. If these terms are a dominant feature, AMG could be an effective preconditioner for a hyperelastic model, when applied to the block diagonal component in a similar approach to the linear elastic model.

Clearly, there is evidence that a block velocity component AMG preconditioning approach is effective when iteratively solving a linear elastic model. Naturally, taking inspiration from the existing literature, this approach could prove to be effective for hyperelastic solids. Further, how would this approach compare when applying both the Stokes-like formulation from the Characteristic-Galerkin discretisation and hyperelastic solid, in conjunction with the FSI formulation? Based on this heuristic from literature, the pursuit of using this block-diagonal AMG approach with a hyperelastic solid is a creditable candidate to be investigated experimentally. It is worth noting that if both the fluid and solid contributions result in separate clusters of eigenvalues, this could lead to poor convergence properties.

The velocity block can be further decomposed into sub-blocks based upon the components of the velocity. With this structure, the action of each diagonal block can be considered independently. This can be computed by applying a fixed number of AMG cycles to each elliptic diagonal block, thus providing a controllable and inexpensive approximation to the action of the inverse velocity block. This is illustrated abstractly as,

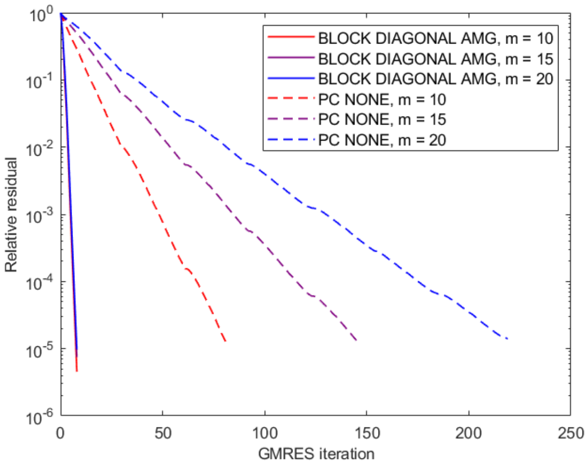
$$\boldsymbol{M}_A^{-1} = \begin{bmatrix} \mathrm{AMG}(A_{xx}) & 0 & 0 \\ 0 & \mathrm{AMG}(A_{yy}) & 0 \\ 0 & 0 & \mathrm{AMG}(A_{zz}) \end{bmatrix}, \tag{6.17}$$

where $\boldsymbol{M}_A^{-1}$ is an approximation of the inverse of the diagonal velocity blocks used for the preconditioner and $\mathrm{AMG}(A_{ii})$ is a single V-cycle of algebraic multigrid cycles.

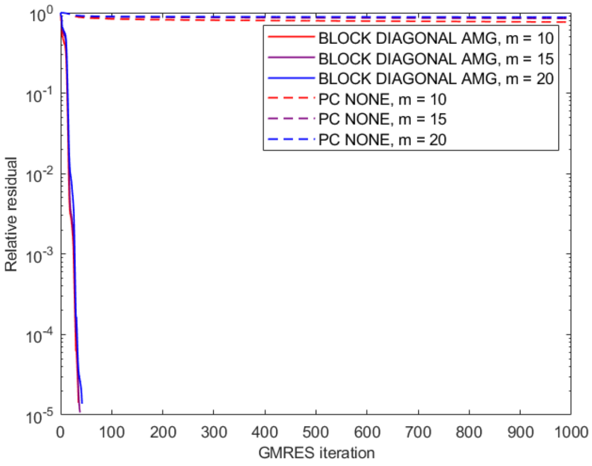
To test the performance of this proposed block preconditioner acting on the velocity block in isolation, the velocity block is solved albeit with a right-hand side set to one,

$$\boldsymbol{A}\boldsymbol{u} = \{1\}_{N_u}^T, \tag{6.18}$$

where the right hand side, $\{1\}_{N_u}^T$, is a column vector of ones, the same dimension of the velocity block $\boldsymbol{A}$. The velocity block is composed of either a pure fluid problem or an FSI problem for comparison.



(a) Fluid only.



(b) Fluid structure interaction.

Figure 6.2: Relative residual of the velocity block solved with restarted GMRES, where m relative number of increments for each mesh resolution.

The geometric and material setup is identical to that in Section 5.2.1. These tests only consider the finite element space P_2/P_1 in three dimensions, where the size of the problem for each mesh resolution is given in Table 6.1.

Figure 6.2a compares the convergence rate when solving just the velocity block for a pure fluid problem (in the absence of a solid) using restarted GMRES, both for the non-preconditioned, and a block preconditioned, solve. The GMRES restart value is 30 iterations, the default choice within PETSc. All subsequent results using GMRES in this Section will use the same restart value, unless stated otherwise. In Figure 6.2b is the same experiment also with an additional contribution from the solid.

m	Total degrees of freedom	Fluid		FSI	
		None	Block AMG	None	Block AMG
10	65441	81	8	1000	35
15	194821	145	8	1000	38
20	439770	219	8	1000	42

Table 6.1: Solution to the velocity block only, number of GMRES iterations taken to converge to a relative tolerance $||r||/||r_0||= 1 \times 10^{-5}$, and maximum number of iterations $itmax = 1000$.

Table 6.1 shows the number of iterations to convergence for three size of problems. In both cases, just fluid and FSI, the application of the preconditioner significantly reduces the number of iterations to convergence. In fact, the preconditioned fluid system performs extremely well, with a fixed number of 8 iterations to convergence, independent of the size of the problem.

Without preconditioning, the FSI system performed significantly worse relative to the fluid-only problem. In Figure 6.2b, GMRES has an extremely slow convergence rate, leading to stall and ultimately not converging within the 1000 iteration limit. With the block-diagonal precondition we observed a significant acceleration in convergence rate, however unlike the fluid-only case it is problem-size dependent.

Even with a large magnitude in the stiffness contribution from the solid acting only in a small portion of the domain, the AMG performs well for the velocity FSI formulation. This demonstrates the robustness of the AMG approach when used to approximate the inverse of the velocity block.

m	Fluid	FSI	Ratio (fluid to FSI)
10	1.906	2.753	0.692
15	4.185	12.736	0.329
20	11.620	36.538	0.308

Table 6.2: Time to solve (in seconds), solved using diagonal block preconditioned GMRES.

m	Fluid		FSI	
	None	Block AMG	None	Block AMG
10	0.0085	0.2382	0.0085	0.0787
15	0.0275	0.5231	0.0272	0.3352
20	0.0645	1.4525	0.0640	0.8700

Table 6.3: Time to solve per iteration (in seconds), solving using GMRES.

Table 6.2 is the total solution times to form and apply the preconditioner and then solve the velocity block system. The preconditioned fluid system does outperform the FSI system in terms of solution time. As the size of the problem increases, the performance of this preconditioner when applied to the FSI system deteriorates relative to that of the fluid system. When we examine the solution time per iteration, as shown in Table 6.3, the preconditioned FSI is faster than the fluid system. This is attributable to the setup cost of the preconditioner being distributed over a greater number of iterations. It is worth noting that for cases without preconditioning, the time per iteration is almost identical, as expected.

6.3.3 Schur complement

The Schur complement arises from the Gaussian elimination of a 3 by 3 linearised system of equations. As discussed previously, it is often used in the preconditioning of block systems.

Exact Schur complement

Calculating the exact Schur complement as a preconditioner will provide optimal convergence when applied to the Krylov subspace iterative methods. The ideal block preconditioner would take the form,

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{A}^{-1} & 0 & 0 \\ 0 & S_p^{-1} & 0 \\ 0 & 0 & S_\lambda^{-1} \end{bmatrix} \tag{6.19}$$

where the exact Schur complement is $S_p = -B\boldsymbol{A}^{-1}B^T$, which is identical to the Schur complement which arises from the factorisation of a 2 by 2 system [147]. Additionally, due to the mean pressure constraint, another Schur complement type block appears in the form of, $S_\lambda = -\boldsymbol{v}_p S_p^{-1} \boldsymbol{v}_p^T$.

Clearly this is impractical; not only does this requires the inverse of $\boldsymbol{A}$ for the velocity block, furthermore another inverse of $\boldsymbol{A}$ is required to calculate the exact Schur complement. Even if the sparse matrix $\boldsymbol{A}$ is sufficiently small, such that the exact inversion is feasible, the inverse would be dense; thus, the exact Schur complement would also become a dense matrix. The inverse of the exact Schur complement, applied as a preconditioner, would now have a significant computational expense. Therefore, it is usually impractical and inefficient to compute an exact Schur complement.

Inexact Schur complement

As a result of the high computational cost of calculating the exact Schur complement, even for smaller problems, it is necessary to formulate a cheaper approximation.

For a close approximation to the Schur complement, using the Moore-Penrose pseudo-inverse of the rectangular matrices applied to the off-diagonal velocity-pressure coupling blocks, $B^\dagger = B^T(BB^T)^{-1}$,

$$\hat{S}^{-1} = (BB^T)^{-1} B\boldsymbol{A}B^T (BB^T)^{-1}, \tag{6.20}$$

has been proposed by Benzi et al. [24]. Although this approach was found to perform poorly in some cases, it was found to be improved by scaling this Schur approximation by two suitable diagonal matrices, such as taking the diagonal of the velocity mass matrix Elman et al. [65]. Approximating a Schur complement is a challenge in itself, however, it is the action of the inverse of this approximation that is required for preconditioning. This approach approximates the inverse Schur complement directly; however, it relies on BB^T being sufficiently sparse that the inverse is cheap to determine.

For purely fluid cases, the pressure mass matrix is a good preconditioner to approximate the exact Schur complement for a discrete Stokes problem [66]. For Navier-Stokes, it is possible to demonstrate spectral equivalence with the full dense matrix $B\boldsymbol{A}^{-1}B^T$ [191]. It is a competitive approximation for low Reynolds flows, however it does not account for the convection contribution to the Schur complement [66]. Since the pressure mass matrix is sparse, the pressure Schur complement approximation is also kept sparse, and due to its well-conditioned structure, it is efficient to solve approximately.

In the FSI formulation there are now also contributions from the solid term, thus this pressure mass matrix is no longer a good approximation to the Schur complement, which has been tested experimentally. There is still a need to find a cost-effective approximation to the inverse of the velocity block, $\boldsymbol{A}$, within the Schur complement. To ensure that the Schur complement is sparse, the approximation of the inverse of $\boldsymbol{A}$ must also be sparse. Thus, a practical choice would be to consider just the diagonal of $\boldsymbol{A}$ [148],

$$\tilde{S} = -B^T \operatorname{diag}(\boldsymbol{A})^{-1} B. \tag{6.21}$$

Here, the inverse of $\operatorname{diag}(\boldsymbol{A})$ is simply the reciprocal of the diagonal entries of the velocity matrix and is cheap to calculate, only requiring N_u operations.

In this case, the approximation of the Schur complement should retain sparsity if B is sparse. Another benefit of using just $\operatorname{diag}(\boldsymbol{A})$, regardless of whether $\boldsymbol{A}$ is symmetric, is that this produces a symmetric Schur complement approximation. Therefore, to approximate the inverse of this approximate Schur complement five iterations of conjugate gradient are taken

$$\tilde{S}^{-1} \approx \mathrm{CG5}(\tilde{S}), \tag{6.22}$$

similar to the approach commonly adopted for the mass matrix [66]. This fixes the cost to form this approximate inverse Schur complement.

The Lambda-Schur complement, S_λ, as outlined in (6.15), reduces to a single scalar value. In the context of the design of the preconditioner for this system, the cost of generating and then inverting S_p to assemble S_λ would be high. Therefore, it would be significantly cheaper to simply substitute the Lambda-Schur complement with a predetermined fixed scalar, for the experiments in this chapter it is set to a value of one.

6.3.4 Enriched pressure space

Extending the analysis from Section 6.3.1 on Taylor-Hood elements, we now consider an enriched pressure space, $P_1 + P_0$, which can be decomposed into a continuous and discontinuous component. The linear system takes the form,

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_{P_1}^T & \boldsymbol{B}_{P_0}^T & 0 \\ \boldsymbol{B}_{P_1} & 0 & 0 & \boldsymbol{v}_{P_1}^T \\ \boldsymbol{B}_{P_0} & 0 & 0 & \boldsymbol{v}_{P_0}^T \\ 0 & \boldsymbol{v}_{P_1} & \boldsymbol{v}_{P_0} & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u} \\ p_{P_1} \\ p_{P_0} \\ \lambda \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{6.23}$$

which includes a mean pressure constraint, constraining both pressures through $\int p^{P_1} + p^{P_0} = 0$.

The system is now in terms of four unknowns increasing the complexity of the factorisation,

$$\mathcal{A} = \mathcal{LDU} = \begin{bmatrix} I & 0 & 0 & 0 \\ l_{10} & I & 0 & 0 \\ l_{20} & l_{21} & I & 0 \\ l_{30} & l_{31} & l_{32} & I \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & 0 & 0 & 0 \\ 0 & S_{P_1} & 0 & 0 \\ 0 & 0 & S_{P_0} & 0 \\ 0 & 0 & 0 & S_\lambda \end{bmatrix} \begin{bmatrix} I & u_{01} & u_{02} & u_{03} \\ 0 & I & u_{12} & u_{13} \\ 0 & 0 & I & u_{23} \\ 0 & 0 & 0 & I \end{bmatrix} \tag{6.24}$$

with the following strict upper triangular terms,

$$\begin{aligned} u_{01} &= \boldsymbol{A}^{-1} B_{P_1}^T \\ u_{02} &= \boldsymbol{A}^{-1} B_{P_0}^T \\ u_{03} &= 0 \\ u_{12} &= S_{P_1}^{-1} B_{P_1} \boldsymbol{A}^{-1} B_{P_0}^T \\ u_{13} &= S_{P_1}^{-1} \boldsymbol{v}_P^T \\ u_{23} &= S_{P_0}^{-1} (\boldsymbol{v}_{P_0}^T + B_{P_0} \boldsymbol{A}^{-1} B_{P_1}^T S_{P_1}^{-1} \boldsymbol{v}_{P_1}^T) \end{aligned}$$

and lower strict triangular terms,

$$\begin{aligned}
l_{10} &= B_{P_1}\boldsymbol{A}^{-1} \\
l_{20} &= B_{P_0}\boldsymbol{A}^{-1} \\
l_{21} &= -B_{P_0}\boldsymbol{A}^{-1}B_{P_1}^T S_{P_1}^{-1} \\
l_{30} &= 0 \\
l_{31} &= \boldsymbol{v}_{P_1} S_{P_1}^{-1} \\
l_{32} &= (\boldsymbol{v}_{P_0} - \boldsymbol{v}_{P_1} S_{P_1}^{-1} B_{P_1}\boldsymbol{A}^{-1}B_{P_0}^T) S_{P_0}^{-1}.
\end{aligned}$$

where a number of different Schur complements arise,

$$\begin{aligned}
S_{P_1} &= -B_{P_1}\boldsymbol{A}^{-1}B_{P_1}^T \\
S_{P_0} &= -B_{P_0}\boldsymbol{A}^{-1}B_{P_0}^T + B_{P_0}\boldsymbol{A}^{-1}B_{P_1}^T S_{P_1}^{-1} B_{P_1}\boldsymbol{A}^{-1}B_{P_0}^T \\
S_\lambda &= \boldsymbol{v}_{P_1} S_{P_1}^{-1}\boldsymbol{v}_{P_1}^T - (\boldsymbol{v}_{P_0} - \boldsymbol{v}_{P_1} S_{P_1}^{-1} B_{P_1}\boldsymbol{A}^{-1}B_{P_0}^T) S_{P_0}^{-1} (\boldsymbol{v}_{P_0}^T + B_{P_0}\boldsymbol{A}^{-1}B_{P_1}^T S_{P_1}^{-1}\boldsymbol{v}_{P_1}^T)
\end{aligned}$$

Given that every term above requires at least a single inversion of $\boldsymbol{A}$, the cost of performing this factorisation exactly would be impractical.

In a similar sense to (6.15) both gradient operator blocks from each pressure space are lumped together, as $B^* = [B_{P_1} B_{P_0}]$. Thus the decomposition becomes,

$$\mathcal{A} = \mathcal{L}\mathcal{D}\mathcal{U} = \begin{bmatrix} I & 0 & 0 \\ -B^*\boldsymbol{A}^{-1} & I & 0 \\ 0 & -\boldsymbol{v}_p S_p^{-1} & I \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & 0 & 0 \\ 0 & S_p^* & 0 \\ 0 & 0 & S_\lambda \end{bmatrix} \begin{bmatrix} I & -\boldsymbol{A}^{-1}B^{*T} & 0 \\ 0 & I & -S_p^{-1}\boldsymbol{v}_p^T \\ 0 & 0 & I \end{bmatrix}, \tag{6.25}$$

where the pressure Schur complement becomes $S_p^* = -B^*\boldsymbol{A}^{-1}B^{*T}$. Since B^* is constructed from the concatenation of two submatrices that will have an irregular sparsity pattern, this may lead to a poorly conditioned system and slow convergence. It is the diagonal matrix, $\mathcal{D}$, of (6.25) that will be used as the structure of the preconditioner from the enriched pressure elements, $P_3/(P_1 + P_0)$.

To conclude this section, a brief summary of the structure of the block preconditioner for each finite element pair is discussed. For the Taylor-Hood element, P_2/P_1, takes the form of (6.16). With an approximation of the velocity block using the AMG of the diagonal blocks, in (6.17), and an approximation of the Schur complement using (6.22) and (6.21). This is also the same preconditioner that will be used for P_3/P_0. Although the enriched pressure space, $P_3/(P_1 + P_0)$, could present a more challenging preconditioning structure, as shown in (6.24), it is simplified by using the diagonal factor, $\mathcal{D}$, of (6.25). The same structure of the preconditioner as in (6.16).

6.4 Numerical results

In this section, the block preconditioning strategy outlined in previous sections is tested with the three-dimensional FSI test case involving the channel wall, as previously described in Section 5.2.2. The block preconditioning approach consists of approximation of the block diagonal velocity components using (6.17) and approximation of the Schur complement with (6.21) and (6.22).

Due to the nature of the ALE algorithm, the mesh movement solution (of the linear elastic equation) is sensitive to the solution of the velocity of the FSI system, which can lead to mesh quality and mesh tangling issues for large deflections of soft solids. This is further compounded by the high mesh density for finer resolutions, hence for the tests in this chapter we choose a stiff solid based upon a solid material constant of $c_1 = 1 \times 10^8$. Even still both the FSI and the mesh movement systems are solved, and there is still a large contribution from the solid term in the FSI system to test the preconditioning approach.

The outer solver of the FSI system uses a GMRES, with the block preconditioner. The block preconditioner consists of a fixed number of algebraic multigrid cycles applied to the block diagonals of the velocity components in (6.17) and a fixed number of iterations of the conjugate gradient, in (6.22), applied to approximate the inverse of the Schur complement (6.21). This approach should provide a cost-effective calculation of a preconditioner while retaining some of the properties of the original system. For the mesh movement, again the linear elastic model is solved using a GMRES with a single cycle of AMG of the entire discrete elastic formulation acting as the preconditioner. The condition for convergence is a relative tolerance of the preconditioned residual of $rtol = 1 \times 10^{-8}$. If the iteration count reaches a maximum of $itmax = 1000$, the solver is terminated. The simulation is run at a stop time, TT. This is summarised in Figure 6.3, which shows a schematic of the solution approach to apply preconditioners and solve system equations for this ALE FSI approach.

The preconditioner for the FSI system will be applied at each iteration of the GMRES.

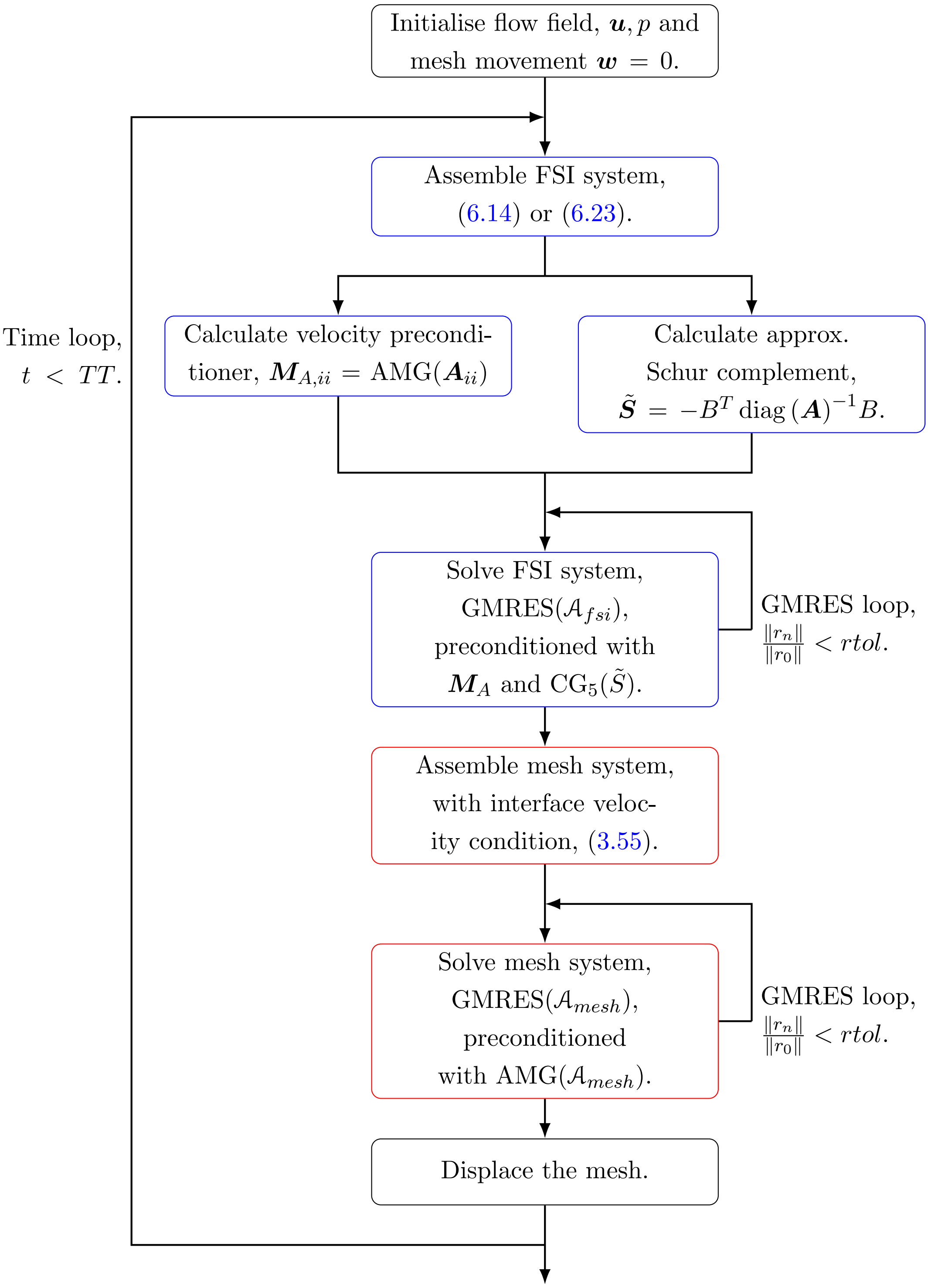


Figure 6.3: Cycle of the assembly of the system and preconditioner, with solve order. Blue boxes indicate the FSI system and red boxes indicate the linear elastic system of the mesh movement.

6.4.1 Direct solve

For small problems, even in three dimensions, a sparse direct solver can be used. In this case, the direct LU solver within PETSc is used. To reduce the bandwidth of the matrix, a reverse Cuthill-McKee sorting algorithm is applied [46]. This aims to reduce fill-in during the factorisation, to provide numerical stability.

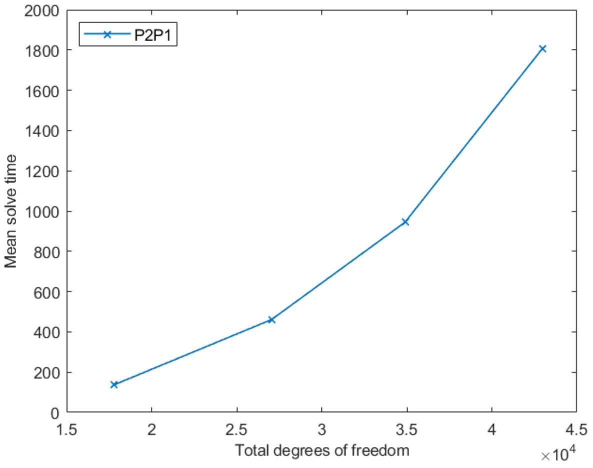


Figure 6.4: LU direct solve, mean solve time averaged over all time steps.

Figure 6.4 is the solution time, mean averaged over all time steps, increasing with the total number of degrees of freedom of the FSI problem. The growth rate of the solve time scales with $N^{3.13}$, where N is the total number of degrees of freedom. For any given piece of hardware, the number of floating-point operations per second (FLOPS) should remain fairly constant; therefore, the time to solve is a good indicator of the number of numerical operations performed. Clearly, a setup with a higher resolution, or high order of FE space, will require even more time and at some point an infeasible amount of memory to solve using direct methods.

When considering discontinuous pressure spaces, P_3/P_0 and $P_3/(P_1 + P_0)$, it is clear that the higher order velocity space generates matrices with more degrees of freedom, relative to the P_2/P_1 for a similar mesh resolution. These large, denser matrices, with therefore require more operations to factorise, resulting in a significantly slower solve time.

6.4.2 Incomplete LU preconditioner

To compare the proposed method outlined in this chapter, an incomplete LU factorisation of the entire discrete system is performed, to act as a preconditioner for the Krylov iterative solver. This acts as a standard out-of-the-box preconditioner typically used for iterative methods,

$$\boldsymbol{M}^{-1} = \mathrm{ILU}\left(\begin{bmatrix} \boldsymbol{A} & B^T \\ B & I \end{bmatrix}\right) \tag{6.26}$$

In this case the system is no longer rearranged by velocity components. The preconditioned system is modified, in (6.26), in order to reduce pivoting, and the zero block is replaced by the identity matrix to avoid zero divisions on the diagonal of this block. To perform the ILU, the Euclid library with HYPRE is called within PETSc [110].

For incomplete LU factorisation, a more accurate preconditioner can be generated by permitting additional fill-in of the factors. The level of fill-in, k, dictates that the factor follows the sparsity pattern of $\mathcal{A}^{k+1}$. The tests are performed using a fill-in level of $k = 1$, which generates a denser sparsity pattern of $\mathcal{A}^2$ relative to $\mathcal{A}$.

In PETSc, for sparse matrices, pivoting is never performed to achieve numerical stability, so the nonzero structure is maintained. However in general a system can be reordered to remove zeros on the diagonal, avoiding zero pivots [11]. Instead PETSc aims to reorder the systems to find a low-fill ordering upon factorisation, reducing the bandwidth of the sparse matrix.

It is found that using just an out-of-the-box ILU preconditioner, albeit with a minor modification to the pressure block (replacing with identity block), is not a robust approach. A relative residual norm is used for the convergence metric for the iterative solver, and the initial residual is found to increase with each successive time step and then diverge with each iteration within each time step. To mitigate this affect, the preconditioner is scaled so that the initial l^2 norm of the preconditioned residual is unity at each time step. It is found from numerical experiments of FSI that the residual norm is divergent over each time step, in that the initial residual norm grows in magnitude dramatically between time steps, which then processes to “runaway” after a number of time steps. Across all problem sizes, the simulations terminate prematurely within 12 time steps due to a divergent solution.

From observations, applying a blanket preconditioner to the entire system is not an effective way to precondition these block systems and should be avoided.

6.4.3 Block preconditioner

A block preconditioning approach, which combines the component-wise algebraic multigrid applied to the velocity block, as outlined in Section 6.3.2, and the approximations to the Schur complements, as outlined in Section 6.3.3, is applied to the channel wall FSI case.

The results in Table 6.4 show the statistics of the GMRES (with restart) iterations for each FE space and each mesh resolution. Compared to the test cases using the direct solver, we note that the size of the problems is significantly larger. The mean iteration count shows very little variation with problem size. The two FE spaces that include the discontinuous pressures, P_0, have a marginally larger mean iteration count relative to the P_2/P_1 space.

m	Degrees of Freedom			Iteration count		
	Velocity	Pressure	Total	Min	Max	Mean
12	132066	5822	137889	6	9	6.95
14	181899	7992	189892	6	9	6.8
16	262584	11429	274014	6	9	6.79
18	331158	14651	345810	6	9	6.91

(a) P_2/P_1

m	Degrees of Freedom			Iteration count		
	Velocity	Pressure	Total	Min	Max	Mean
10	277146	19097	296244	8	11	8.94
12	435033	30410	465444	8	10	8.58
14	599823	42017	641841	7	10	7.68
16	826143	57934	884078	8	10	8.51

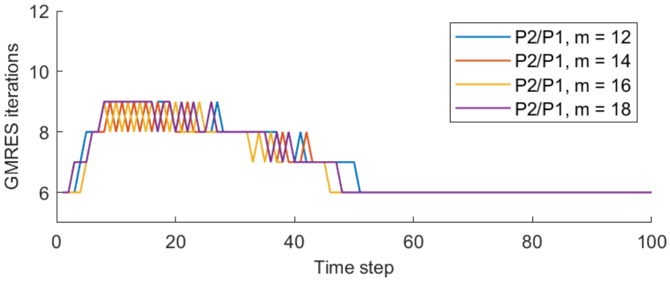
(b) P_3/P_0

m	Degrees of Freedom			Iteration count		
	Velocity	Pressure	Total	Min	Max	Mean
10	277146	22963	300110	8	11	8.82
12	392508	32628	425137	8	10	8.48
14	565293	47081	612375	8	10	8.51
16	826143	69066	895210	8	10	8.54

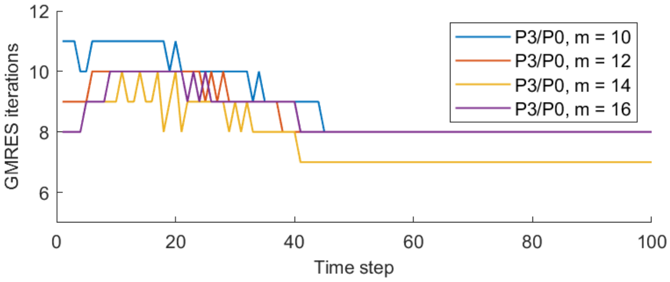
(c) $P_3/(P_1 + P_0)$

Table 6.4: Degree of freedom and GMRES iteration count to convergence for each mesh resolution.

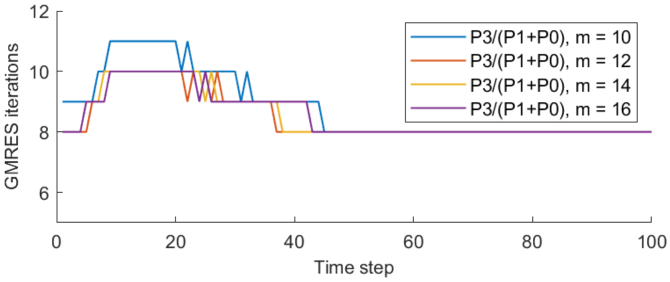
In Figure 6.5 the number of GMRES (with restart) iterations to convergence does vary in time; however it remains bounded independently of the size of the mesh (see Table 6.4 for details of the mesh). For this test problem, the largest iterations counts occur at the early time steps when the solid motion is at its greatest. As the solid reaches a maximum deflection and a steady state, at the midpoint of the simulation (time step 50), the number of iterations also reaches a minimal steady state value. It is observed that the number of GMRES iterations does not vary between the different mesh resolutions, and thus is independent of the problem size.



(a) P_2/P_1



(b) P_3/P_0



(c) $P_3/(P_1 + P_0)$

Figure 6.5: Number of GMRES iterations at each time step. The mesh resolution is indicated by the value of m.

Figure 6.6 shows the mean solve time plotted against the total number of degrees of freedom. It is already observable that this preconditioning approach is not optimal in the sense that the solve time does not grow linearly. The component-wise AMG applied to the velocity block has been shown to be a very good preconditioner by reducing the number of iterations and solve time (when considering the FSI formulation) in isolation.

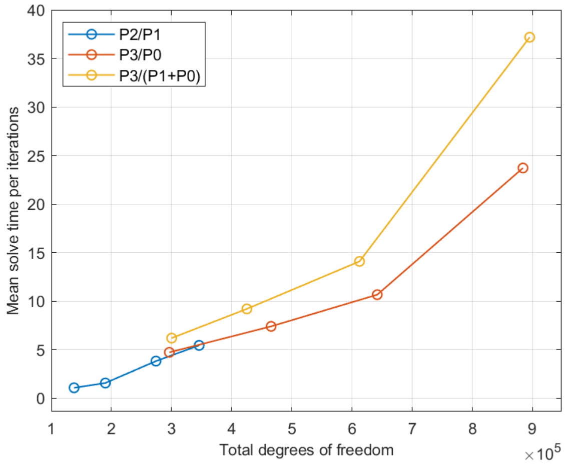


Figure 6.6: Mean solve time per GMRES iterations with total degrees of freedom.

However, considering a fixed iteration count is observed with the size of the problem, in Table 6.4, yet the time to solve grows in a superlinear fashion in Figure 6.6, clearly there is an operation growing nonlinearly. This could be the cost of applying the AMG to the velocity block. Since the AMG will recursively coarsen to the smallest possible problem, within the single V-cycle of the number of hierarchies (depth of cycle) will grow with problem size. Alternatively, calling both packages PETSc and HYPRE within FreeFEM may involve a transfer in the data structures of each package with an associated overhead cost (in time and memory) that generates a bottleneck in the solution algorithm.

To make a more direct comparison between the LU direct solver and GMRES with block preconditioner, the mean solve time at each time is plotted against the total degrees of freedom for P_2/P_1, in Figure 6.7.

Clearly, the block-preconditioned GMRES method outperforms the LU direct solver, solving larger problems in less time. Note that the LU direct solver is limited by the available memory. Similarly, GMRES with the block preconditioner demonstrates better scalability, with problem size, compared to the direct solver, $N^{1.87}$ and $N^{3.13}$ growth rates, respectively. The preconditioned iterative solver, however, does not exhibit an optimal solve rate. When plotted against an exponential best fit ($aN^b + c$), dashed line in Figure 6.7, the block-preconditioned GMRES deviates from this fit. Again, the direct solver is limited in application

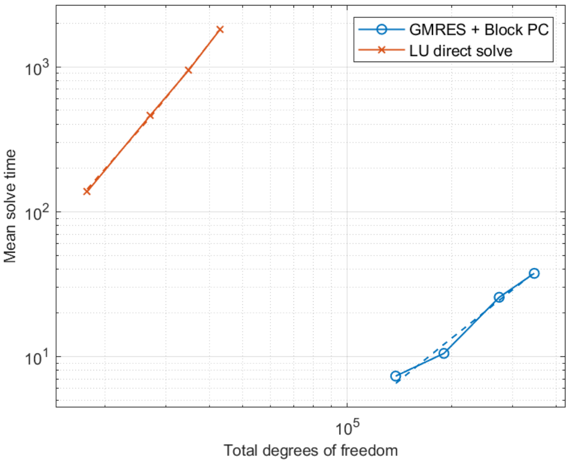


Figure 6.7: Mean solve time at each time step with total degrees of freedom for P_2/P_1 finite element space.

to larger problems due to the memory restrictions of the hardware.

6.5 Concluding remarks

The preconditioner proposed in this chapter, for application to a finite element ALE formulation for modelling FSI, consists of a block-diagonal AMG approximation to the velocity components, as outlined in Section 6.3.2, and the approximation action of an inexact Schur complement, as outlined in Section 6.3.3.

This block-diagonal AMG preconditioner applied to solving the velocity block in isolation, in Section 6.3.2, performs well in reducing the number of iterations to convergence when preconditioning a fluid and FSI problem. In terms of solve time, the block-diagonal preconditioner performs better compared to without a preconditioner, applied to a GMRES iterative solver, as shown in Table 6.2. This is particularly noticeable when applied to the FSI.

When applying the block diagonal AMG preconditioner, with an efficient approximation to the Schur complement, to act as an entire block preconditioner with GMRES, the number of iterations to convergence is found to be independent of the size of the problem (shown in Figure 6.5).

To benchmark this block-diagonal AMG preconditioning approach, the use of ILU as a preconditioner is not sufficiently robust. The block-diagonal AMG preconditioned GMRES performs better than the LU direct solver, both in terms of performance and scalability (with problem size), shown in Figure 6.7. However, it exhibits suboptimal scalability with problem size, $N^{1.87}$, for the P_2/P_1 finite element space. It is suspected that the finite element pairing with cubic velocity would scale even worse, given that they would generate even large problems. Without the use of parallelised solver (alongside high-memory hardware), a solution would not feasible be found.

The use of the unstructured mesh of the tetrahedral element in three dimensions within the ALE framework could also contribute to difficulties in maintaining mesh quality control. Although it does provide benefits in meshing more complex geometries, particularly curved surfaces, when performing the mesh movement, the tetrahedral elements are particularly susceptible to compression or elongation of element. For this reason, it has only been possible to conduct robust tests for problems with relatively small deformations in this chapter. Future work, outside the scope of this thesis, should look at more robust mesh deformation strategies, which would allow a wider variety of test problems to be considered in order to further assess the robustness of the proposed preconditioner.

Chapter 7

Conclusions

This section will conclude the results of the numerical experiments to investigate the performance of applying different pressure finite element spaces in two- and three-dimensional ALE-FSI-FE method (outlined in Chapter 3), referring back to Chapters 4 and 5 respectively. After which, conclusions will be drawn on the use of the block preconditioning approach outlined in Chapter 6.

7.1 Finite element spaces

In two dimensions, when solving with a direct solver, it is the P_2/P_0 finite element space that solves faster given a specific number of degrees of freedom. Although the two pressure spaces with the inclusion of the discontinuous pressure space, P_2/P_0 and $P_2/(P_1+P_0)$ tend to yield results that are in close agreement (relative to those obtained using P_2/P_1 elements). For the test cases considered, the inclusion of the enriched pressure space, $P_2/(P_1+P_0)$, does not provide performance savings over P_2/P_0 when comparing the same problem sizes. In fact, when examining the pressure contribution from both the P_1 and P_0 spaces in the $P_2/(P_1 + P_0)$ space, as shown in Figure 4.11 and Figure 4.21, the dominant component is the discontinuous pressure component.

In practice, the challenges of maintaining a high quality moving mesh mean that the single-mesh ALE approach for FSI is limited to small deflections without the use of discrete remeshing or adaptive mesh refinement. This is particularly problematic for cases that involved oscillatory responses from the filament; for example refer to 2D Test 2, in Section 4.3.2, a filament attached behind a fixed cylinder, where the quality of the mesh deteriorates so that the system becomes so poorly conditioned that the simulation terminates prematurely.

We find that for the stability of finite element spaces in three dimensions, our observations agree with the analysis of Boffi et al. [28]. Particularly for the use of discontinuous pressure finite element space, there is indeed a need to include a midface degree of freedom, to ensure mass continuity between elements. Although the preferred method of Silvester [175] is to use a bubble function to introduce an additional degree of freedom in the middle of the face, we do so using a cubic velocity space since we are uncertain of the precise implementation of the bubble functions in FreeFEM.

For Test 1 in three dimensions, it is P_2/P_1 and $P_3/(P_1 + P_0)$ which provide good agreement in terms of the displacements of the solid, as shown in 5.7. With finite element spaces with discontinuous pressures, P_3/P_0 and $P_3/(P_1 + P_0)$, outperforming P_2/P_1 marginally both in terms of memory and solve time. On the contrary, in Test 2 in three dimensions, it is the finite element spaces with discontinuous pressure spaces, P_3/P_0 and $P_3/(P_1+P_0)$ for which the solid displacements agree.

In both two- and three-dimensional cases, the solution size of the problem is limited by the memory when using direct solvers.

Throughout this project, we have targeted resolution of the pressure-continuity at the solid-fluid interface through the choice of finite element space. It is demonstrated that, for FSI problems, finite elements with a discontinuous pressure component are a preferable choice over the typical continuous pressure finite element, such as the Taylor-Hood element is popular with fluid problems. Alternative approaches could be used to capture the discontinuity of pressure, which for FSI will occur at the interface between the fluid and solid, without the need to use a high mesh resolution. For example, one such method could consider two separate pressures that occupy the constitutive regions, such that $p^f \in \Omega^f$ and $p^s \in \Omega^s$.

7.2 Preconditioning for FSI ALE

The block preconditioning approach considered for this FE-ALE-FSI formulation considers a single V-cycle of AMG applied to the block diagonal, velocity components. To approximate the Schur complement, the diagonal of the velocity block is taken, and a maximum of five iterations of a conjugate gradient iterative method are applied.

The concept of using approximations to component-wise velocity block diagonals using AMG has been demonstrated to be an effective preconditioner for linear elastic problems. The AMG block-diagonal preconditioned GMRES applied the FSI velocity in isolation exhibits significantly better performance than when solving with GMRES without a preconditioner, as shown in Section 6.3.2.

When using this block preconditioning approach in conjunction with GMRES, the number of iterations to convergence between fixed with problem size and pressure space. We observe that this preconditioning combination performs better and scales better than a direct LU solution, as shown in Figure 6.7. Importantly, we also find that other preconditioners, such as ILU, or an unpreconditioned iterative solver, are not sufficiently robust to find a solution to this formulation. When examining the solve time of the block-preconditioner, suboptimal scalability with problem is observed. We attribute this to the growth in the number of hierarchies in the AMG V-cycle, or the communication between packages with FreeFEM, since the number of GMRES iterations is bound.

A key question would be, is this preconditioning approach applicable to other FSI methods? In Chapter 6, the proposed preconditioner exploits the structure of the system. Any discretisation method which forms this saddle point system structure should be appropriate, under the constraint that the velocity block retain properties such that AMG provides an efficient approximation to the inverse of the diagonal blocks.

7.3 Future work

This section outlines future avenues of research in the context of the work presented in this thesis and the ALE FSI formulation.

The solid model tested in the simulations considers only the incompressible neo-Hookean hyperelastic solid, as outlined in Section 3.3.1. The choice of solid models is broad, as briefly discussed in Section 3.3. Further investigation could measure the performance of this preconditioning approach (outline in Section 6.3) with the application of different hyperelastic solid models. Since this preconditioner is designed around the block structure of saddle-point systems, as long as the solid retains the elliptic properties, the component-wise block AMG should provide an efficient approximation.

As shown in Sections 5.2.1 and 5.2.2 the deflections of the solid filament are severely limited in magnitude. To ensure that this ALE method is a more general and robust approach to solving FSI problems with larger deflections, mesh adaptivity in order to maintain mesh quality would need to be further considered. The ALE approach in this project does perform mesh adaptivity, by solving the linear elastic equation for the mesh movement is a form of r-refinement. One method would be to consider a discrete remeshing of the domain when the mesh quality reduces to a certain threshold. However, this will introduce an interpolation error between the meshes. Alternatively, adaptive mesh refinement in the form of h-refinement could provide a continuous approach to maintain mesh quality. This coupled with an efficient iterative solver would permit the application of this approach to a broader range of problems in three dimensions.

All simulations throughout this thesis were performed in serial, on a single processor core. Naturally, a future investigation would examine the scalability of this preconditioning approach on multiple processors. Parallelisation in FreeFEM is implemented using a domain decomposition approach. There would still be some technical challenges on how the preconditioner would be calculated across the different sub-domains.

Bibliography

[1] E. B. Agamloh, A. K. Wallace, and A. von Jouanne. Application of fluidstructure interaction simulation of an ocean wave energy extraction device. *Renewable Energy*, 33(4):748–757, 4 2008. ISSN 0960-1481. doi: 10.1016/J.RENENE.2007.04.010.

[2] P. R. Amestoy, I. S. Duff, and J. Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Computer Methods in Applied Mechanics and Engineering*, 184(2-4):501–520, 4 2000. ISSN 00457825. doi: 10.1016/S0045-7825(99)00242-X.

[3] D. N. Arnold. Mixed finite element methods for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 82(1-3):281–300, 9 1990. ISSN 0045-7825. doi: 10.1016/0045-7825(90)90168-L.

[4] E. M. Arruda and M. C. Boyce. A three-dimensional constitutive model for the large stretch behavior of rubber elastic materials. *Journal of the Mechanics and Physics of Solids*, 41(2):389–412, 1993. ISSN 00225096. doi: 10.1016/0022-5096(93)90013-6.

[5] F. P. T. Baaijens. A fictitious domain/mortar element method for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 35(7):743–761, 4 2001. ISSN 02712091. doi: 10.1002/1097-0363(20010415)35:7⟨743::AID-FLD109⟩3.0.CO;2-A.

[6] I. Babuška. Error-bounds for finite element method. *Numerische Mathematik 1971 16:4*, 16(4):322–333, 1 1971. ISSN 0945-3245. doi:

10.1007/BF02165003. URL https://link.springer.com/article/10.1007/BF02165003.

[7] S. Badia, A. Quaini, and A. Quarteroni. Modular vs. non-modular preconditioners for fluid-structure systems with large added-mass effect. *Computer Methods in Applied Mechanics and Engineering*, 197(49-50):4216–4232, 9 2008. ISSN 00457825. doi: 10.1016/J.CMA.2008.04.018.

[8] S. Badia, A. Quaini, and A. Quarteroni. Splitting Methods Based on Algebraic Factorization for Fluid-Structure Interaction. *https://doi.org/10.1137/070680497*, 30(4):1778–1805, 4 2008. ISSN 10648275. doi: 10.1137/070680497. URL https://epubs.siam.org/doi/10.1137/070680497.

[9] W. Bai and R. Eatock Taylor. Fully nonlinear simulation of wave interaction with fixed and floating flared structures. *Ocean Engineering*, 36(3-4):223–236, 3 2009. ISSN 0029-8018. doi: 10.1016/J.OCEANENG.2008.11.003.

[10] Z. Z. Bai, G. H. Golub, and M. K. Ng. Hermitian and Skew-Hermitian Splitting Methods for Non-Hermitian Positive Definite Linear Systems. *https://doi.org/10.1137/S0895479801395458*, 24(3):603–626, 7 2006. ISSN 08954798. doi: 10.1137/S0895479801395458. URL https://epubs.siam.org/doi/10.1137/S0895479801395458.

[11] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. M. Constantinescu, L. Dalcin, A. Dener, and others. PETSc/TAO Users Manual Revision 3.19. Technical report, Argonne National Lab.(ANL), Argonne, IL (United States), 2023.

[12] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. B. Brune, Peter, K. Buschelman, E. M. Constantinescu, L. Dalcin, A. D. Eijkhout, Victor, J. Faibussowitsch, W. D. Gropp, V. Hapla, T. Isaac, P. J. Karpeev, Dmitry, D. Kaushik, M. G. Knepley, F. Kong, S. K. May, D. A., L. C. McInnes, R. T. Mills, L. Mitchell, T. M. Roman, J. E., K. Rupp, P. Sanan, J. Sarich, B. F. S. Zampini, Stefano, H. Zhang, H. Zhang, and J. Zhang. PETSc Web page. https://petsc.org/, 2023. URL https://petsc.org/.

[13] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics, 1 1994. ISBN 978-0-89871-328-2. doi: 10.1137/1.9781611971538. URL http://epubs.siam.org/doi/book/10.1137/1.9781611971538.

[14] K. J. Bathe and H. Zhang. Finite element developments for general fluid flows with structural interactions. *International Journal for Numerical Methods in Engineering*, 60(1):213–232, 5 2004. ISSN 1097-0207. doi: 10.1002/NME.959. URL https://onlinelibrary.wiley.com/doi/full/10.1002/nme.959.

[15] Y. Bazilevs, V. M. Calo, Y. Zhang, and T. J. Hughes. Isogeometric fluid-structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38(4-5):310–322, 2006. ISSN 01787675. doi: 10.1007/s00466-006-0084-3.

[16] Y. Bazilevs, M. C. Hsu, Y. Zhang, W. Wang, X. Liang, T. Kvamsdal, R. Brekken, and J. G. Isaksen. A fully-coupled fluid-structure interaction simulation of cerebral aneurysms. *Computational Mechanics 2009 46:1*, 46 (1):3–16, 10 2009. ISSN 1432-0924. doi: 10.1007/S00466-009-0421-4. URL https://link.springer.com/article/10.1007/s00466-009-0421-4.

[17] Y. Bazilevs, M. C. Hsu, Y. Zhang, W. Wang, T. Kvamsdal, S. Hentschel, and J. G. Isaksen. Computational vascular fluid-structure interaction: Methodology and application to cerebral aneurysms. *Biomechanics and Modeling in Mechanobiology*, 9(4):481–498, 8 2010. ISSN 16177959. doi: 10.1007/S10237-010-0189-7/METRICS. URL https://link.springer.com/article/10.1007/s10237-010-0189-7.

[18] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger. 3D simulation of wind turbine rotors at full scale. Part II: Fluid-structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65(1-3):236–253, 1 2011. ISSN 02712091. doi: 10.1002/fld.2454. URL http://doi.wiley.com/10.1002/fld.2454.

[19] Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. Challenges and directions in computational fluid-structure interaction. *Mathematical Models and Methods in Applied Sciences*, 23(2):215–221, 2 2013. ISSN 02182025. doi: 10.1142/S0218202513400010.

[20] Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. *Computational Fluid-Structure Interaction: Methods and Applications*. John Wiley & Sons, 2013. ISBN 978-0-470-97877-1. URL `https://www.wiley.com/en-us/Computational+Fluid+Structure+Interaction%3A+Methods+and+Applications-p-9780470978771#:~:text=a%20dummy%20description.-,Description,solution%20of%20real%2Dworld%20problems.`

[21] M. Benzi. Preconditioning Techniques for Large Linear Systems: A Survey. *Journal of Computational Physics*, 182(2):418–477, 11 2002. ISSN 0021-9991. doi: 10.1006/JCPH.2002.7176.

[22] M. Benzi and G. H. Golub. A Preconditioner for Generalized Saddle Point Problems. *SIAM Journal on Matrix Analysis and Applications*, 26(1):20–41, 1 2004. ISSN 0895-4798. doi: 10.1137/S0895479802417106. URL `http://epubs.siam.org/doi/10.1137/S0895479802417106`.

[23] M. Benzi and M. A. Olshanskii. An Augmented LagrangianBased Approach to the Oseen Problem. *SIAM Journal on Scientific Computing*, 28(6):2095–2113, 1 2006. ISSN 1064-8275. doi: 10.1137/050646421. URL `http://epubs.siam.org/doi/10.1137/050646421`.

[24] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 5 2005. ISSN 0962-4929. doi: 10.1017/S0962492904000212. URL `https://www.cambridge.org/core/product/identifier/S0962492904000212/type/journal_article`.

[25] D. Boffi and L. Gastaldi. A fictitious domain approach with Lagrange multiplier for fluid-structure interactions. *Numerische Mathematik 2016 135:3*, 135(3):711–732, 6 2016. ISSN 0945-3245. doi: 10.1007/S00211-016-0814-1. URL `https://link.springer.com/article/10.1007/s00211-016-0814-1`.

[26] D. Boffi, F. Brezzi, and M. Fortin. Finite Elements for the Stokes Problem. In *Lecture Notes in Mathematics*, volume 1939, pages 45–100. 2008. doi: 10.1007/978-3-540-78319-0_2. URL http://link.springer.com/10.1007/978-3-540-78319-0_2.

[27] D. Boffi, N. Cavallini, F. Gardini, and L. Gastaldi. Immersed boundary method: Performance analysis of popular finite element spaces. In *Proceedings of the 4th International Conference on Computational Methods for Coupled Problems in Science and Engineering, COUPLED PROBLEMS 2011*, pages 135–146, 2011. ISBN 9788489925786.

[28] D. Boffi, N. Cavallini, F. Gardini, and L. Gastaldi. Local mass conservation of stokes finite elements. *Journal of Scientific Computing*, 52(2):383–400, 2012. ISSN 08857474. doi: 10.1007/s10915-011-9549-4.

[29] J. Bonet and R. D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*, volume 9780521838702. Cambridge University Press, 3 2008. ISBN 9780521838702. doi: 10.1017/CBO9780511755446. URL https://www.cambridge.org/core/product/identifier/9780511755446/type/book.

[30] A. F. Bower. *Applied Mechanics of Solids.* CRC Press, 10 2009. ISBN 9781439802489. doi: 10.1201/9781439802489. URL https://www.taylorfrancis.com/books/9781439802489.

[31] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics.* Springer New York, New York, NY, 2008. ISBN 978-0-387-75933-3. doi: 10.1007/978-0-387-75934-0. URL http://link.springer.com/10.1007/978-0-387-75934-0.

[32] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 8(R2):129–151, 5 1974. ISSN 0397-9342. doi: 10.1051/m2an/

197408R201291. URL http://www.esaim-m2an.org/10.1051/m2an/197408R201291.

[33] W. L. Briggs, V. E. Henson, and S. F. McCormick. A Multigrid Tutorial, Second Edition. *A Multigrid Tutorial, Second Edition*, 1 2000. doi: 10.1137/1.9780898719505.

[34] J. Brown. Efficient nonlinear solvers for nodal high-order finite elements in 3D. *Journal of Scientific Computing*, 45(1-3):48–63, 10 2010. ISSN 08857474. doi: 10.1007/S10915-010-9396-8/METRICS. URL https://link.springer.com/article/10.1007/s10915-010-9396-8.

[35] V. Bruyere, N. Fillot, G. E. Morales-Espejel, and P. Vergne. Computational fluid dynamics and full elasticity model for sliding line thermal elastohydro dynamic contacts. *Tribology International*, 46(1):3–13, 2012. ISSN 0301679X. doi: 10.1016/J.TRIBOINT.2011.04.013.

[36] E. Burman and M. A. Fernández. An unfitted Nitsche method for incompressible fluid-structure interaction using overlapping meshes. *Computer Methods in Applied Mechanics and Engineering*, 279:497–514, 9 2014. ISSN 00457825. doi: 10.1016/J.CMA.2014.07.007.

[37] A. Calderer, X. Guo, L. Shen, and F. Sotiropoulos. Fluidstructure interaction simulation of floating structures interacting with complex, large-scale ocean waves and atmospheric turbulence with application to floating offshore wind turbines. *Journal of Computational Physics*, 355:144–175, 2 2018. ISSN 00219991. doi: 10.1016/j.jcp.2017.11.006. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999117308331.

[38] J. O. Campos, R. W. dos Santos, J. Sundnes, and B. M. Rocha. Preconditioned augmented Lagrangian formulation for nearly incompressible cardiac mechanics. *International Journal for Numerical Methods in Biomedical Engineering*, 34(4):e2948, 4 2018. ISSN 2040-7947. doi: 10.1002/CNM.2948. URL https://onlinelibrary.wiley.com/doi/full/10.1002/cnm.2948.

[39] P. Causin, J. F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering*, 194(42-44):4506–4527, 10 2005. ISSN 00457825. doi: 10.1016/J.CMA.2004.12.005.

[40] E. Celledoni, G. Johannessen, and T. Kvamsdal. Parallelisation of a CFD code: the use of Aztec library in the parallel numerical simulation of Extrusion of Aluminium. In *Parallel Computational Fluid Dynamics 2000*, pages 291–298. Elsevier, 1 2001. doi: 10.1016/B978-044450673-3/50104-4. URL `https://linkinghub.elsevier.com/retrieve/pii/B9780444506733501044`.

[41] U. Chaduvula, D. Patel, and N. Gopalakrishnan. Fluid-Structure-Soil Interaction Effects on Seismic Behaviour of Elevated Water Tanks. *Procedia Engineering*, 51:84–91, 1 2013. ISSN 1877-7058. doi: 10.1016/J.PROENG.2013.01.014.

[42] S. Chen and G. D. Doolen. Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30(1):329–364, 1 1998. ISSN 0066-4189. doi: 10.1146/annurev.fluid.30.1.329. URL `https://www.annualreviews.org/doi/10.1146/annurev.fluid.30.1.329`.

[43] C.-Y. Chiang, O. Pironneau, T. Sheu, and M. Thiriet. Numerical Study of a 3D Eulerian Monolithic Formulation for Incompressible Fluid-Structures Systems. *Fluids*, 2(2):34, 6 2017. ISSN 2311-5521. doi: 10.3390/fluids2020034. URL `http://www.mdpi.com/2311-5521/2/2/34`.

[44] T. J. Chung. Introduction. In *Computational Fluid Dynamics*, pages 3–28. Cambridge University Press, 2 2002. doi: 10.1017/CBO9780511606205.003. URL `https://www.cambridge.org/core/product/identifier/CBO9780511606205A007/type/book_part`.

[45] A. Cunningham, Jr. The role of non-linear aerodynamics in fluid-structure interaction. In *29th AIAA, Fluid Dynamics Conference*, pages 1–14, Reston, Virigina, 6 1998. American Institute of Aeronautics and Astronautics.

doi: 10.2514/6.1998-2423. URL https://arc.aiaa.org/doi/10.2514/6.1998-2423.

[46] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. *Proceedings of the 1969 24th National Conference, ACM 1969*, pages 157–172, 8 1969. doi: 10.1145/800195.805928. URL https://dl.acm.org/doi/10.1145/800195.805928.

[47] H. Dal, A. Kemal, and Y. Badienia. On the Performance of Isotropic Hyperelastic Constitutive Models for Rubber-Like Materials: A State of the Art Review. *Applied Mechanics Reviews*, 73(2), 3 2021. ISSN 00036900. doi: 10.1115/1.4050978/1108153. URL https://asmedigitalcollection.asme.org/appliedmechanicsreviews/article/73/2/020802/1108153/On-the-Performance-of-Isotropic-Hyperelastic.

[48] T. A. Davis. Algorithm 832: UMFPACK V4.3 - An unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2): 196–199, 6 2004. ISSN 00983500. doi: 10.1145/992200.992206.

[49] A. de Boer, M. S. van der Schoot, and H. Bijl. Mesh deformation based on radial basis function interpolation. *Computers and Structures*, 85(11-14): 784–795, 6 2007. ISSN 00457949. doi: 10.1016/J.COMPSTRUC.2007.01.013.

[50] E. de Langre. Effects of Wind on Plants. *Annual Review of Fluid Mechanics*, 40(1):141–168, 1 2008. ISSN 0066-4189. doi: 10.1146/annurev.fluid.40.111406.102135. URL https://www.annualreviews.org/doi/10.1146/annurev.fluid.40.111406.102135.

[51] E. de Sturler and J. Liesen. Block-Diagonal and Constraint Preconditioners for Nonsymmetric Indefinite Linear Systems. Part I: Theory. *SIAM Journal on Scientific Computing*, 26(5):1598–1619, 1 2005. ISSN 1064-8275. doi: 10.1137/S1064827502411006. URL http://epubs.siam.org/doi/10.1137/S1064827502411006.

[52] M. D. De Tullio, A. Cristallo, E. Balaras, and R. Verzicco. Direct numerical simulation of the pulsatile flow through an aortic bileaflet mechanical heart valve. *Journal of Fluid Mechanics*, 622:259–290, 3 2009. ISSN 0022-1120. doi: 10.1017/S0022112008005156. URL `https://www.cambridge.org/core/product/identifier/S0022112008005156/type/journal_article`.

[53] C. Degand and C. Farhat. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computers and Structures*, 80(3-4):305–316, 2 2002. ISSN 00457949. doi: 10.1016/S0045-7949(02)00002-0.

[54] J. Degroote and J. Vierendeels. Multi-solver algorithms for the partitioned simulation of fluidstructure interaction. *Computer Methods in Applied Mechanics and Engineering*, 200(25-28):2195–2210, 6 2011. ISSN 0045-7825. doi: 10.1016/J.CMA.2011.03.015.

[55] J. Degroote, K. J. Bathe, and J. Vierendeels. Performance of a new partitioned procedure versus a monolithic procedure in fluidstructure interaction. *Computers & Structures*, 87(11-12):793–801, 6 2009. ISSN 0045-7949. doi: 10.1016/J.COMPSTRUC.2008.11.013.

[56] S. Deparis, D. Forti, G. Grandperrin, and A. Quarteroni. FaCSI: A block parallel preconditioner for fluidstructure interaction in hemodynamics. *Journal of Computational Physics*, 327:700–718, 12 2016. ISSN 00219991. doi: 10.1016/j.jcp.2016.10.005. URL `https://linkinghub.elsevier.com/retrieve/pii/S0021999116304983`.

[57] J. Donea, S. Giuliani, and J. P. Halleux. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33(1-3):689–723, 9 1982. ISSN 0045-7825. doi: 10.1016/0045-7825(82)90128-1.

[58] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian Methods. In *Encyclopedia of Computational Mechanics*. John Wiley & Sons, Ltd, Chichester, UK, 11 2004. doi:

10.1002/0470091355.ecm009. URL https://onlinelibrary.wiley.com/doi/10.1002/0470091355.ecm009.

[59] D. Dowson. Paper 12: Modes of Lubrication in Human Joints. *Proceedings of the Institution of Mechanical Engineers, Conference Proceedings*, 181 (10):45–54, 6 1966. ISSN 0367-8849. doi: 10.1243/PIME_CONF_1966_181_206_02. URL http://journals.sagepub.com/doi/10.1243/PIME_CONF_1966_181_206_02.

[60] I. S. Duff. Direct Methods. Technical report, Rutherford Appleton Laboratory, Oxon, 7 1998.

[61] I. S. Duff and J. K. Reid. The Multifrontal Solution of Indefinite Sparse Symmetric Linear. *ACM Transactions on Mathematical Software*, 9(3): 302–325, 9 1983. ISSN 0098-3500. doi: 10.1145/356044.356047. URL https://dl.acm.org/doi/10.1145/356044.356047.

[62] I. S. Duff, A. M. Erisman, and J. K. Reid. Direct Methods for Sparse Matrices. *Direct Methods for Sparse Matrices*, 4 2017. doi: 10.1093/ACPROF:OSO/9780198508380.001.0001. URL https://academic.oup.com/book/27908.

[63] S. Dupont, F. Gosselin, C. Py, E. De Langre, P. Hemon, and Y. Brunet. Modelling waving crops using large-eddy simulation: comparison with experiments and a linear stability analysis. *Journal of Fluid Mechanics*, 652:5–44, 2010. ISSN 1469-7645. doi: 10.1017/S0022112010000686. URL https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/modelling-waving-crops-using-largeeddy-simulation-comparison-with-experimen
5362233D71E2866819F611F9C14607CE.

[64] H. Elman and D. Silvester. Fast Nonsymmetric Iterations and Preconditioning for Navier-Stokes Equations. *SIAM Journal on Scientific Computing*, 17(1):33–46, 1 1996. ISSN 1064-8275. doi: 10.1137/0917004. URL http://epubs.siam.org/doi/10.1137/0917004.

[65] H. Elman, V. E. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro. Block preconditioners based on approximate commutators. *SIAM Journal on Scientific Computing*, 27(5):1651–1668, 2006. ISSN 10648275. doi: 10.1137/040608817. URL http://www.siam.org/journals/sisc/27-5/60881.html.

[66] H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers*, volume 53. Oxford University Press, 6 2014. ISBN 9780199678792. doi: 10.1093/acprof:oso/9780199678792.001.0001. URL https://oxford.universitypressscholarship.com/view/10.1093/acprof:oso/9780199678792.001.0001/acprof-9780199678792.

[67] H. C. Elman. Preconditioning for the Steady-State Navier–Stokes Equations with Low Viscosity. *SIAM Journal on Scientific Computing*, 20(4):1299–1316, 1 1999. ISSN 1064-8275. doi: 10.1137/S1064827596312547. URL http://epubs.siam.org/doi/10.1137/S1064827596312547.

[68] M. Eswaran, U. Saha, and D. Maity. Effect of baffles on a partially filled cubic tank: Numerical simulation and experimental validation. *Computers & Structures*, 87(3-4):198–205, 2 2009. ISSN 00457949. doi: 10.1016/j.compstruc.2008.10.008. URL https://linkinghub.elsevier.com/retrieve/pii/S0045794908002149.

[69] R. D. Falgout and U. M. Yang. hypre: A library of high performance preconditioners. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2331 LNCS(PART 3):632–641, 2002. ISSN 16113349. doi: 10.1007/3-540-47789-6_66/COVER. URL https://link.springer.com/chapter/10.1007/3-540-47789-6_66.

[70] C. Farhat, M. Lesoinne, and P. LeTallec. 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(1-2):95–114, 4 1998. ISSN 0045-7825. doi: 10.1016/S0045-7825(97)00216-8.

[71] L. J. Fauci and R. Dillon. Biofluidmechanics of reproduction, 12 2006. ISSN 00664189. URL https://www.annualreviews.org/doi/abs/10.1146/annurev.fluid.37.061903.175725.

[72] M. A. Fernández and M. Moubachir. A Newton method using exact jacobians for solving fluid-structure coupling. In *Computers and Structures*, volume 83, pages 127–142. Elsevier Ltd, 2005. doi: 10.1016/j.compstruc.2004.04.021.

[73] M. A. Fernández, J. F. Gerbeau, and C. Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *International Journal for Numerical Methods in Engineering*, 69(4):794–821, 1 2007. ISSN 1097-0207. doi: 10.1002/NME.1792. URL https://onlinelibrary.wiley.com/doi/full/10.1002/nme.1792.

[74] B. Fischer, A. Ramage, D. J. Silvester, and A. J. Wathen. Minimum residual methods for augmented systems. *BIT Numerical Mathematics 1998 38:3*, 38(3):527–543, 1998. ISSN 1572-9125. doi: 10.1007/BF02510258. URL https://link.springer.com/article/10.1007/BF02510258.

[75] R. Fletcher. Conjugate gradient methods for indefinite systems. pages 73–89, 1976. doi: 10.1007/BFB0080116. URL https://link.springer.com/chapter/10.1007/BFb0080116.

[76] L. A. Freitag and C. Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. *International Journal for Numerical Methods in Engineering*, 40(21):3979–4002, 11 1997. ISSN 0029-5981. doi: 10.1002/(SICI)1097-0207(19971115)40:21⟨3979::AID-NME251⟩3.0.CO;2-9. URL https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1097-0207(19971115)40:21<3979::AID-NME251>3.0.CO;2-9.

[77] R. W. Freund and N. M. Nachtigal. QMR: a quasi-minimal residual method for non-Hermitian linear systems. *Numerische Mathematik*, 60(1):315–339, 12 1991. ISSN 0029599X. doi: 10.1007/BF01385726/METRICS. URL https://link.springer.com/article/10.1007/BF01385726.

[78] A. Gauthier, F. Saleri, and A. Veneziani. A fast preconditioner for the incompressible Navier Stokes Equations. *Computing and Visualization in Science*, 6(2-3):105–112, 3 2004. ISSN 1432-9360. doi: 10.1007/s00791-003-0114-z. URL http://link.springer.com/10.1007/s00791-003-0114-z.

[79] L. Ge and F. Sotiropoulos. A Numerical Method for Solving the 3D Unsteady Incompressible Navier-Stokes Equations in Curvilinear Domains with Complex Immersed Boundaries. *Journal of computational physics*, 225(2):1782, 8 2007. ISSN 10902716. doi: 10.1016/J.JCP.2007.02.017. URL /pmc/articles/PMC2635106//pmc/articles/PMC2635106/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC2635106/.

[80] M. W. Gee, U. Küttler, and W. A. Wall. Truly monolithic algebraic multigrid for fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 85(8):987–1016, 2 2011. ISSN 00295981. doi: 10.1002/nme.3001. URL https://onlinelibrary.wiley.com/doi/10.1002/nme.3001.

[81] A. N. Gent. A New Constitutive Relation for Rubber. *Rubber Chemistry and Technology*, 69(1):59–61, 3 1996. ISSN 0035-9475. doi: 10.5254/1.3538357. URL https://meridian.allenpress.com/rct/article/69/1/59/92301/A-New-Constitutive-Relation-for-Rubber.

[82] R. Glowinski, T. W. Pan, T. I. Hesla, and D. D. Joseph. A distributed Lagrange multiplier/fictitious domain method for particulate flows. *International Journal of Multiphase Flow*, 25(5):755–794, 8 1999. ISSN 0301-9322. doi: 10.1016/S0301-9322(98)00048-2.

[83] R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, and J. Périaux. A Fictitious Domain Approach to the Direct Numerical Simulation of Incompressible Viscous Flow past Moving Rigid Bodies: Application to Particulate Flow. *Journal of Computational Physics*, 169(2):363–426, 5 2001. ISSN 00219991. doi: 10.1006/jcph.2000.6542.

[84] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4th edition, 2013. ISBN 9781421407944. URL `https://jhupbooks.press.jhu.edu/title/matrix-computations`.

[85] A. Greenbaum. Iterative Methods for Solving Linear Systems. *Iterative Methods for Solving Linear Systems*, 1 1997. doi: 10.1137/1.9781611970937.

[86] C. Greif and D. Schotzau. Preconditioners for saddle point linear systems with highly singular (1,1) blocks. *Electronic Transactions on Numerical Analysis*, 22:114–121, 2006. ISSN 1068-9613.

[87] P. M. Gresho, R. L. Sani, and M. S. Engelman. *Incompressible flow and the finite element method*, volume 2. John Wiley and Sons, 7 2000. ISBN 978-0-471-49250-4.

[88] M. Griebel, T. Neunhoeffer, and H. Regler. Algebraic multigrid methods for the solution of the Navier-Stokes equations in complicated geometries. *International Journal for Numerical Methods in Fluids*, 26(3):281–301, 2 1998. ISSN 0271-2091. doi: 10.1002/(SICI)1097-0363(19980215)26:3⟨281::AID-FLD632⟩3.0.CO;2-2. URL `https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1097-0363(19980215)26:3<281::AID-FLD632>3.0.CO;2-2`.

[89] B. E. Griffith and X. Luo. Hybrid finite difference/finite element immersed boundary method. *International Journal for Numerical Methods in Biomedical Engineering*, 33(12):e2888, 12 2017. ISSN 2040-7939. doi: 10.1002/cnm.2888. URL `https://onlinelibrary.wiley.com/doi/10.1002/cnm.2888`.

[90] B. E. Griffith, X. Luo, D. M. McQueen, and C. S. Peskin. Simulating the fluid dynamics of natural and prosthetic heart valves using the immersed boundary method. *International Journal of Applied Mechanics*, 01(01):137–177, 3 2009. ISSN 1758-8251. doi: 10.1142/S1758825109000113. URL `https://www.worldscientific.com/doi/abs/10.1142/S1758825109000113`.

[91] J. Han and C. S. Peskin. Spontaneous oscillation and fluidstructure interaction of cilia. *Proceedings of the National Academy of Sciences*, 115(17): 4417–4422, 4 2018. ISSN 0027-8424. doi: 10.1073/pnas.1712042115. URL `https://pnas.org/doi/full/10.1073/pnas.1712042115`.

[92] P. Hansbo, J. Hermansson, and T. Svedberg. Nitsche's method combined with spacetime finite elements for ALE fluidstructure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193(39-41):4195–4206, 10 2004. ISSN 00457825. doi: 10.1016/j.cma.2003.09.029. URL `https://linkinghub.elsevier.com/retrieve/pii/S004578250400218X`.

[93] F. Hecht. New development in freefem++. *Journal of Numerical Mathematics*, 20(3-4):251–265, 1 2012. ISSN 1569-3953. doi: 10.1515/jnum-2012-0013. URL `https://www.degruyter.com/document/doi/10.1515/jnum-2012-0013/html`.

[94] F. Hecht and O. Pironneau. An energy stable monolithic Eulerian fluid-structure finite element method. *International Journal for Numerical Methods in Fluids*, 85(7):430–446, 11 2017. ISSN 10970363. doi: 10.1002/FLD.4388.

[95] M. Heil. An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193(1-2):1–23, 1 2004. ISSN 00457825. doi: 10.1016/j.cma.2003.09.006.

[96] M. Heil, A. L. Hazel, and J. Boyle. Solvers for large-displacement fluid-structure interaction problems: Segregated versus monolithic approaches. *Computational Mechanics*, 43(1):91–101, 2008. ISSN 01787675. doi: 10.1007/s00466-008-0270-6.

[97] V. E. Henson and U. M. Yang. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41(1):155–177, 4 2002. ISSN 01689274. doi: 10.1016/S0168-9274(01)00115-5. URL `https://linkinghub.elsevier.com/retrieve/pii/S0168927401001155`.

[98] C. Hesch, A. J. Gil, A. Arranz Carreño, J. Bonet, and P. Betsch. A mortar approach for FluidStructure interaction problems: Immersed strategies for deformable and rigid bodies. *Computer Methods in Applied Mechanics and Engineering*, 278:853–882, 8 2014. ISSN 0045-7825. doi: 10.1016/J.CMA.2014.06.004.

[99] A. Hessenthaler, N. R. Gaddum, O. Holub, R. Sinkus, O. Röhrle, and D. Nordsletten. Experiment for validation of fluid-structure interaction models and algorithms. *International Journal for Numerical Methods in Biomedical Engineering*, 33(9):e2848, 9 2017. ISSN 2040-7947. doi: 10.1002/CNM.2848. URL `https://onlinelibrary.wiley.com/doi/full/10.1002/cnm.2848`.

[100] M. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409, 12 1952. ISSN 0091-0635. doi: 10.6028/jres.049.044. URL `https://nvlpubs.nist.gov/nistpubs/jres/049/jresv49n6p409_A1b.pdf`.

[101] C. Hirt, A. Amsden, and J. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253, 3 1974. ISSN 00219991. doi: 10.1016/0021-9991(74)90051-5. URL `https://linkinghub.elsevier.com/retrieve/pii/0021999174900515`.

[102] G. Hou, J. Wang, and A. Layton. Numerical methods for fluid-structure interaction - A review, 8 2012. ISSN 18152406. URL `https://www.cambridge.org/core/journals/communications-in-computational-physics/article/numerical-methods-for-fluidstructure-interaction-a-review/359AE654882EAFB08CAAA862AC7B05AC`.

[103] J. S. Hou, M. H. Holmes, W. M. Lai, and V. C. Mow. Boundary Conditions at the Cartilage-Synovial Fluid Interface for Joint Lubrication and Theoretical Verifications. *Journal of Biomechanical Engineering*, 111(1):78–87, 2 1989. ISSN 0148-0731. doi: 10.1115/1.3168343. URL `https:`

//asmedigitalcollection.asme.org/biomechanical/article/111/1/78/396811/Boundary-Conditions-at-the-CartilageSynovial-Fluid.

[104] M.-C. Hsu and Y. Bazilevs. Fluidstructure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics*, 50(6): 821–833, 12 2012. ISSN 0178-7675. doi: 10.1007/s00466-012-0772-0. URL http://link.springer.com/10.1007/s00466-012-0772-0.

[105] M. C. Hsu, D. Kamensky, Y. Bazilevs, M. S. Sacks, and T. J. Hughes. Fluidstructure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. *Computational Mechanics*, 54(4):1055–1071, 10 2014. ISSN 01787675. doi: 10.1007/S00466-014-1059-4/FIGURES/7. URL https://link.springer.com/article/10.1007/s00466-014-1059-4.

[106] D. Z. Huang, P. Avery, C. Farhat, J. Rabinovitch, A. Derkevorkian, and L. D. Peterson. Modeling, Simulation and Validation of Supersonic Parachute Inflation Dynamics during Mars Landing. In *AIAA Scitech 2020 Forum*, Reston, Virginia, 1 2020. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-595-1. doi: 10.2514/6.2020-0313. URL https://arc.aiaa.org/doi/10.2514/6.2020-0313.

[107] B. Hübner, E. Walhorn, and D. Dinkler. A monolithic approach to fluid-structure interaction using spacetime finite elements. *Computer Methods in Applied Mechanics and Engineering*, 193(23-26):2087–2104, 6 2004. ISSN 0045-7825. doi: 10.1016/J.CMA.2004.01.024.

[108] B. Hübner, W. Weber, and U. Seidel. The role of fluid-structure interaction for safety and life time prediction in hydraulic machinery. *IOP Conference Series: Earth and Environmental Science*, 49(7):072007, 11 2016. ISSN 1755-1307. doi: 10.1088/1755-1315/49/7/072007. URL https://iopscience.iop.org/article/10.1088/1755-1315/49/7/072007.

[109] T. J. Hughes, W. K. Liu, and T. K. Zimmermann. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29(3):329–349, 12 1981.

ISSN 00457825. doi: 10.1016/0045-7825(81)90049-9. URL https://linkinghub.elsevier.com/retrieve/pii/0045782581900499.

[110] D. Hysom and A. Pothen. A Scalable Parallel Algorithm for Incomplete Factor Preconditioning. *SIAM Journal on Scientific Computing*, 22(6): 2194–2215, 1 2001. ISSN 1064-8275. doi: 10.1137/S1064827500376193. URL http://epubs.siam.org/doi/10.1137/S1064827500376193.

[111] S. Ikeda, T. Yamada, and Y. Toda. Numerical study on turbulent flow and honami in and above flexible plant canopy. *International Journal of Heat and Fluid Flow*, 22(3):252–258, 6 2001. ISSN 0142727X. doi: 10.1016/S0142-727X(01)00087-X.

[112] D. Jodlbauer, U. Langer, and T. Wick. Parallel block-preconditioned monolithic solvers for fluid-structure interaction problems. *International Journal for Numerical Methods in Engineering*, 117(6):623–643, 2 2019. ISSN 00295981. doi: 10.1002/nme.5970. URL http://doi.wiley.com/10.1002/nme.5970.

[113] A. Johnson and T. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 119(1-2): 73–94, 11 1994. ISSN 00457825. doi: 10.1016/0045-7825(94)00077-8. URL https://linkinghub.elsevier.com/retrieve/pii/0045782594000778.

[114] C. Kadapa, W. G. Dettmer, and D. Perić. A fictitious domain/distributed Lagrange multiplier based fluidstructure interaction scheme with hierarchical B-Spline grids. *Computer Methods in Applied Mechanics and Engineering*, 301:1–27, 4 2016. ISSN 0045-7825. doi: 10.1016/J.CMA.2015.12.023.

[115] V. Kalro and T. E. Tezduyar. A parallel 3D computational method for fluid-structure interactions in parachute systems. *Computer Methods in Applied Mechanics and Engineering*, 190(3-4):321–332, 10 2000. ISSN 0045-7825. doi: 10.1016/S0045-7825(00)00204-8.

[116] D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, and T. J. Hughes. An immersogeometric variational framework for fluidstructure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053, 2 2015. ISSN 00457825. doi: 10.1016/j.cma.2014.10.040. URL https://linkinghub.elsevier.com/retrieve/pii/S0045782514004101.

[117] D. C. Kellermann and M. M. Attard. An invariant-free formulation of neo-Hookean hyperelasticity. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 96(2):233–252, 2 2016. ISSN 1521-4001. doi: 10.1002/ZAMM.201400210. URL https://onlinelibrary.wiley.com/doi/full/10.1002/zamm.201400210.

[118] J. Kim, D. Kim, and H. Choi. An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries. *Journal of Computational Physics*, 171(1):132–150, 7 2001. ISSN 0021-9991. doi: 10.1006/JCPH.2001.6778.

[119] D. Knoll and D. Keyes. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 1 2004. ISSN 00219991. doi: 10.1016/j.jcp.2003.08.010. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999103004340.

[120] R. J. Knops and L. E. Payne. *Uniqueness Theorems in Linear Elasticity*, volume 19 of *Springer Tracts in Natural Philosophy*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1971. ISBN 978-3-642-65103-8. doi: 10.1007/978-3-642-65101-4. URL http://link.springer.com/10.1007/978-3-642-65101-4.

[121] F. Kong and X. C. Cai. A scalable nonlinear fluidstructure interaction solver based on a Schwarz preconditioner with isogeometric unstructured coarse spaces in 3D. *Journal of Computational Physics*, 340:498–518, 7 2017. ISSN 10902716. doi: 10.1016/J.JCP.2017.03.043.

[122] R. Kral and E. Kreuzer. Multibody Systems and Fluid-Structure Interactions with Application to Floating Structures. *Multibody System Dynamics 1999 3:1*, 3(1):65–83, 1999. ISSN 1573-272X. doi: 10.1023/A:1009710901886. URL `https://link.springer.com/article/10.1023/A:1009710901886`.

[123] O. Ladyzhenskaya. *The Mathematical Theory of Viscous Incompressible Flow*. Gordon & Breach, London, 2. edition, 1969.

[124] I. S. Lan, J. Liu, W. Yang, and A. L. Marsden. A reduced unified continuum formulation for vascular fluidstructure interaction. *Computer Methods in Applied Mechanics and Engineering*, 394, 5 2022. ISSN 00457825. doi: 10.1016/J.CMA.2022.114852.

[125] U. Langer and H. Yang. Partitioned solution algorithms for fluidstructure interaction problems with hyperelastic models. *Journal of Computational and Applied Mathematics*, 276:47–61, 3 2015. ISSN 0377-0427. doi: 10.1016/J.CAM.2014.08.007.

[126] P. Le Tallec and J. Mouro. Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3039–3067, 3 2001. ISSN 00457825. doi: 10.1016/S0045-7825(00)00381-9.

[127] C.-Y. Liaw and A. K. Chopra. Dynamics of towers surrounded by water. *Earthquake Engineering & Structural Dynamics*, 3(1):33–49, 1974. ISSN 00988847. doi: 10.1002/eqe.4290030104. URL `https://onlinelibrary.wiley.com/doi/10.1002/eqe.4290030104`.

[128] J. Liu, W. Yang, M. Dong, and A. L. Marsden. The nested block preconditioning technique for the incompressible Navier-Stokes equations with emphasis on hemodynamic simulations. *Computer Methods in Applied Mechanics and Engineering*, 367:113122, 8 2020. ISSN 00457825. doi: 10.1016/j.cma.2020.113122. URL `https://linkinghub.elsevier.com/retrieve/pii/S0045782520303078`.

[129] M. Liu and Z. Zhang. Smoothed particle hydrodynamics (SPH) for modeling fluid-structure interactions. *Science China Physics, Mechanics & Astronomy 2019 62:8*, 62(8):1–38, 3 2019. ISSN 1869-1927. doi: 10.1007/S11433-018-9357-0. URL `https://link.springer.com/article/10.1007/s11433-018-9357-0`.

[130] R. Löhner and C. Yang. Improved ALE mesh velocities for moving bodies. *Communications in Numerical Methods in Engineering*, 12(10):599–608, 10 1996. ISSN 10698299. doi: 10.1002/(SICI)1099-0887(199610)12:10⟨599::AID-CNM1⟩3.0.CO;2-Q. URL `https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1099-0887(199610)12:10<599::AID-CNM1>3.0.CO;2-Q`.

[131] R. D. Lonsdale. An algebraic multigrid solver for the navier-stokes equations on unstructured meshes. *International Journal of Numerical Methods for Heat & Fluid Flow*, 3(1):3–14, 1 1993. ISSN 09615539. doi: 10.1108/EB017512/FULL/XML.

[132] Q. W. Ma, G. X. Wu, and R. Eatock Taylor. Finite element simulation of fully non-linear interaction between vertical cylinders and steep waves. Part 1: Methodology and numerical procedure. *International Journal for Numerical Methods in Fluids*, 36(3):265–285, 6 2001. ISSN 02712091. doi: 10.1002/FLD.131.

[133] J. R. Magnus and J. R. Magnus. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley Series in Probability and Statistics. Wiley, 2 2019. ISBN 9781119541202. doi: 10.1002/9781119541219. URL `https://onlinelibrary.wiley.com/doi/book/10.1002/9781119541219`.

[134] P. J. Martínez-Ferrer, L. Qian, Z. Ma, D. M. Causon, and C. G. Mingham. An efficient finite-volume method to study the interaction of two-phase fluid flows with elastic structures. *Journal of Fluids and Structures*, 83:54–71, 11 2018. ISSN 08899746. doi: 10.1016/j.jfluidstructs.2018.08.019. URL `https://linkinghub.elsevier.com/retrieve/pii/S0889974618302755`.

[135] H. G. Matthies and J. Steindorf. Partitioned strong coupling algorithms for fluidstructure interaction. *Computers & Structures*, 81(8-11):805–812, 5 2003. ISSN 0045-7949. doi: 10.1016/S0045-7949(02)00409-1.

[136] M. Mayr, M. H. Noll, and M. W. Gee. A hybrid interface preconditioner for monolithic fluid-structure interaction solvers. *Adv. Model. and Simul. in Eng. Sci*, 7:15, 2020. doi: 10.1186/s40323-020-00150-9. URL `https://doi.org/10.1186/s40323-020-00150-9`.

[137] J. B. Medley, D. Dowson, and V. Wright. Transient Elastohydrodynamic Lubrication Models for the Human Ankle Joint. *Engineering in Medicine*, 13(3):137–151, 7 1984. ISSN 0046-2039. doi: 10.1243/EMED_JOUR_1984_013_035_02. URL `http://journals.sagepub.com/doi/10.1243/EMED_JOUR_1984_013_035_02`.

[138] A. Mehraban, J. Brown, V. Barra, H. Tufo, J. Thompson, and R. Regueiro. Efficient Residual and Matrix-Free Jacobian Evaluation for Three-Dimensional Tri-Quadratic Hexahedral Finite Elements With Nearly-Incompressible Neo-Hookean Hyperelasticity Applied to Soft Materials on Unstructured Meshes in Parallel, With PETSc and libCEED. *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*, 12, 2 2021. doi: 10.1115/IMECE2020-24522. URL `https://dx.doi.org/10.1115/IMECE2020-24522`.

[139] C. Michler, E. H. van Brummelen, S. J. Hulshoff, and R. de Borst. The relevance of conservation for stability and accuracy of numerical methods for fluidstructure interaction. *Computer Methods in Applied Mechanics and Engineering*, 192(37-38):4195–4215, 9 2003. ISSN 0045-7825. doi: 10.1016/S0045-7825(03)00392-X.

[140] C. Michler, S. J. Hulshoff, E. H. van Brummelen, and R. de Borst. A monolithic approach to fluid-structure interaction. *Computers and Fluids*, 33(5-6):839–848, 2004. ISSN 00457930. doi: 10.1016/j.compfluid.2003.06.006.

[141] M. D. Mihajlović and S. Mijalković. A component decomposition preconditioning for 3D stress analysis problems. *Numerical Linear Algebra with Applications*, 9(6-7):567–583, 9 2002. ISSN 1070-5325. doi: 10.1002/nla.298. URL `https://onlinelibrary.wiley.com/doi/10.1002/nla.298`.

[142] R. Mittal and G. Iaccarino. IMMERSED BOUNDARY METHODS. *Annual Review of Fluid Mechanics*, 37(1):239–261, 1 2005. ISSN 0066-4189. doi: 10.1146/annurev.fluid.37.061903.175743. URL `https://www.annualreviews.org/doi/10.1146/annurev.fluid.37.061903.175743`.

[143] M. Mooney. A Theory of Large Elastic Deformation. *Journal of Applied Physics*, 11(9):582–592, 9 1940. ISSN 0021-8979. doi: 10.1063/1.1712836. URL `/aip/jap/article/11/9/582/138391/A-Theory-of-Large-Elastic-Deformation`.

[144] Y. Mori and C. S. Peskin. Implicit second-order immersed boundary methods with boundary mass. *Computer Methods in Applied Mechanics and Engineering*, 197(25-28):2049–2067, 4 2008. ISSN 0045-7825. doi: 10.1016/J.CMA.2007.05.028.

[145] V. C. Mow and W. M. Lai. Recent Developments in Synovial Joint Biomechanics. *https://doi.org/10.1137/1022056*, 22(3):275–317, 8 2006. ISSN 0036-1445. doi: 10.1137/1022056. URL `https://epubs.siam.org/doi/10.1137/1022056`.

[146] R. L. Muddle, M. Mihajlović, and M. Heil. An efficient preconditioner for monolithically-coupled large-displacement fluidstructure interaction problems with pseudo-solid mesh updates. *Journal of Computational Physics*, 231(21):7315–7334, 8 2012. ISSN 0021-9991. doi: 10.1016/J.JCP.2012.07.001.

[147] M. F. Murphy, G. H. Golub, and A. J. Wathen. A Note on Preconditioning for Indefinite Linear Systems. *SIAM Journal on Scientific Computing*, 21(6):1969–1972, 1 2000. ISSN 1064-8275. doi: 10.1137/S1064827599355153. URL `http://epubs.siam.org/doi/10.1137/S1064827599355153`.

[148] M. Neytcheva. On element-by-element Schur complement approximations. *Linear Algebra and its Applications*, 434(11):2308–2324, 6 2011. ISSN 0024-3795. doi: 10.1016/J.LAA.2010.03.031.

[149] J. T. Oden and N. Kikuchi. Finite element methods for constrained problems in elasticity. *International Journal for Numerical Methods in Engineering*, 18(5):701–725, 5 1982. ISSN 1097-0207. doi: 10.1002/NME.1620180507. URL https://onlinelibrary.wiley.com/doi/full/10.1002/nme.1620180507.

[150] R. W. Ogden. Large deformation isotropic elasticity on the correlation of theory and experiment for incompressible rubberlike solids. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 326(1567):565–584, 2 1972. ISSN 0080-4630. doi: 10.1098/rspa.1972.0026. URL https://royalsocietypublishing.org/doi/10.1098/rspa.1972.0026.

[151] R. W. Ogden. *Non-Linear Elastic Deformations.* 1997. ISBN 9780486696485.

[152] C. C. Paige and M. A. Saunders. Solution of Sparse Indefinite Systems of Linear Equations. *https://doi.org/10.1137/0712047*, 12(4):617–629, 7 2006. ISSN 00361429. doi: 10.1137/0712047. URL https://epubs.siam.org/doi/10.1137/0712047.

[153] N. R. Paulson, F. Sadeghi, and W. Habchi. A coupled finite element EHL and continuum damage mechanics model for rolling contact fatigue. *Tribology International*, 107:173–183, 3 2017. ISSN 0301679X. doi: 10.1016/J.TRIBOINT.2016.11.024.

[154] M. Pernice and M. D. Tocci. A multigrid-preconditioned Newton-Krylov method for the incompressible Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 23(2):398–418, 2002. ISSN 10648275. doi: 10.1137/S1064827500372250. URL https://epubs.siam.org/terms-privacy.

[155] C. S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, 10 1972. ISSN 00219991. doi:

10.1016/0021-9991(72)90065-4. URL https://linkinghub.elsevier.com/retrieve/pii/0021999172900654.

[156] G. Pichon, M. Faverge, P. Ramet, and J. Roman. Reordering Strategy for Blocking Optimization in Sparse Linear Solvers. *SIAM Journal on Matrix Analysis and Applications*, 38(1):226–248, 1 2017. ISSN 0895-4798. doi: 10.1137/16M1062454. URL http://epubs.siam.org/doi/10.1137/16M1062454.

[157] S. Piperno and C. Farhat. Partitioned procedures for the transient solution of coupled aeroelastic problems - part II: Energy transfer analysis and three-dimensional applications. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3147–3170, 3 2001. ISSN 00457825. doi: 10.1016/S0045-7825(00)00386-8.

[158] O. Pironneau. On the transport-diffusion algorithm and its applications to the Navier-Stokes equations. *Numerische Mathematik 1982 38:3*, 38 (3):309–332, 10 1982. ISSN 0945-3245. doi: 10.1007/BF01396435. URL https://link.springer.com/article/10.1007/BF01396435.

[159] O. Pironneau and M. Tabata. Stability and convergence of a Galerkin-characteristics finite element scheme of lumped mass type. *International Journal for Numerical Methods in Fluids*, 64(10-12):1240–1253, 12 2010. ISSN 02712091. doi: 10.1002/fld.2459. URL https://onlinelibrary.wiley.com/doi/10.1002/fld.2459.

[160] O. Pironneau, J. Liou, and T. Tezduyar. Characteristic-Galerkin and Galerkin/least-squares space-time formulations for the advection-diffusion equation with time-dependent domains. *Computer Methods in Applied Mechanics and Engineering*, 100:117–141, 1992.

[161] G. Popov, S. Sankar, T. Sankar, and G. Vatistas. Liquid sloshing in rectangular road containers. *Computers & Fluids*, 21(4):551–569, 10 1992. ISSN 00457930. doi: 10.1016/0045-7930(92)90006-H. URL https://linkinghub.elsevier.com/retrieve/pii/004579309290006H.

[162] M. Razzaq, H. Damanik, J. Hron, A. Ouazzi, and S. Turek. FEM multigrid techniques for fluid-structure interaction with application to hemodynamics. *Applied Numerical Mathematics*, 62(9):1156–1170, 9 2012. ISSN 01689274. doi: 10.1016/J.APNUM.2010.12.010.

[163] S. Rebouillat and D. Liksonov. Fluidstructure interaction in partially filled liquid containers: A comparative review of numerical approaches. *Computers & Fluids*, 39(5):739–746, 5 2010. ISSN 0045-7930. doi: 10.1016/J.COMPFLUID.2009.12.010.

[164] T. Richter. A monolithic geometric multigrid solver for fluid-structure interactions in ALE formulation. *International Journal for Numerical Methods in Engineering*, 104(5):372–390, 11 2015. ISSN 10970207. doi: 10.1002/NME.4943.

[165] T. Richter and T. Wick. Finite elements for fluid-structure interaction in ALE and fully Eulerian coordinates. *Computer Methods in Applied Mechanics and Engineering*, 199(41-44):2633–2642, 10 2010. ISSN 00457825. doi: 10.1016/j.cma.2010.04.016.

[166] J. W. Ruge and K. Stüben. 4. Algebraic Multigrid. In *Multigrid Methods*, pages 73–130. Society for Industrial and Applied Mathematics, 1 1987. doi: 10.1137/1.9781611971057.ch4. URL http://epubs.siam.org/doi/10.1137/1.9781611971057.ch4.

[167] S. Rugonyi and K. Bathe. On Finite Element Analysis of Fluid Flows Fully Coupled with Structural Interactions. *Computer Modeling in Engineering & Sciences*, 2(2):195–212, 1 1970. ISSN 1526-1492. doi: 10.3970/cmes.2001.002.195. URL https://www.techscience.com/CMES/v2n2/24728/html.

[168] K. J. Ruschak. A method for incorporating free boundaries with surface tension in finite element fluid-flow simulators. *International Journal for Numerical Methods in Engineering*, 15(5):639–648, 5 1980. ISSN 0029-5981. doi: 10.1002/nme.1620150502. URL https://onlinelibrary.wiley.com/doi/10.1002/nme.1620150502.

[169] T. Rusten and R. Winther. A Preconditioned Iterative Method for Saddlepoint Problems. *SIAM Journal on Matrix Analysis and Applications*, 13(3):887–904, 7 1992. ISSN 0895-4798. doi: 10.1137/0613054. URL `http://epubs.siam.org/doi/10.1137/0613054`.

[170] Y. Saad. ILUT: A dual threshold incomplete LU factorization. *Numerical Linear Algebra with Applications*, 1(4):387–402, 7 1994. ISSN 1070-5325. doi: 10.1002/nla.1680010405. URL `https://onlinelibrary.wiley.com/doi/10.1002/nla.1680010405`.

[171] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2nd edition, 1 2003. ISBN 978-0-89871-534-7. doi: 10.1137/1.9780898718003. URL `http://epubs.siam.org/doi/book/10.1137/1.9780898718003`.

[172] Y. Saad and M. H. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 7 1986. ISSN 0196-5204. doi: 10.1137/0907058. URL `http://epubs.siam.org/doi/10.1137/0907058`.

[173] L. A. Sangalli and A. L. Braun. A fluid-structure interaction model for numerical simulation of bridge flutter using sectional models with active control devices. Preliminary results. *Journal of Sound and Vibration*, 477, 7 2020. ISSN 10958568. doi: 10.1016/J.JSV.2020.115338.

[174] H. Si. TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. *ACM Transactions on Mathematical Software (TOMS)*, 41(2), 2 2015. ISSN 15577295. doi: 10.1145/2629697. URL `https://dl.acm.org/doi/10.1145/2629697`.

[175] D. Silvester. Private Communication: Email with David Silvester, 7 2022.

[176] K. Singh, F. Sadeghi, T. Russell, S. J. Lorenz, W. Peterson, J. Villarreal, and T. Jinmon. FluidStructure Interaction Modeling of Elastohydrodynamically Lubricated Line Contacts. *Journal of Tribology*, 143(9), 9 2021. ISSN 0742-4787. doi: 10.1115/1.4049260.

URL https://asmedigitalcollection.asme.org/tribology/article/143/9/091602/1091936/Fluid-Structure-Interaction-Modeling-of.

[177] A. Slone, K. Pericleous, C. Bailey, and M. Cross. Dynamic fluidstructure interaction using finite volume unstructured mesh procedures. *Computers & Structures*, 80(5-6):371–390, 3 2002. ISSN 00457949. doi: 10.1016/S0045-7949(01)00177-8. URL https://linkinghub.elsevier.com/retrieve/pii/S0045794901001778.

[178] K. Stein, R. Benney, V. Kalro, T. E. Tezduyar, J. Leonard, and M. Accorsi. Parachute fluidstructure interactions: 3-D computation. *Computer Methods in Applied Mechanics and Engineering*, 190(3-4):373–386, 10 2000. ISSN 0045-7825. doi: 10.1016/S0045-7825(00)00208-5.

[179] R. Stenberg. Analysis of Mixed Finite Element Methods for the Stokes Problem: A Unified Approach. *Mathematics of Computation*, 42(165):9, 1 1984. ISSN 00255718. doi: 10.2307/2007557. URL https://www.jstor.org/stable/2007557?origin=crossref.

[180] R. Stenberg. On some three-dimensional finite elements for incompressible media. *Computer Methods in Applied Mechanics and Engineering*, 63(3): 261–269, 8 1987. ISSN 00457825. doi: 10.1016/0045-7825(87)90072-7. URL https://linkinghub.elsevier.com/retrieve/pii/0045782587900727.

[181] K. Stüben. Algebraic Multigrid (AMG) : An Introduction With Applications. Technical report, 1999.

[182] K. Stüben. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, 128(1-2):281–309, 3 2001. ISSN 0377-0427. doi: 10.1016/S0377-0427(00)00516-1.

[183] K. Stuben. Appendix A An Introduction to Algebraic Multigrid. In *Multigrid*, pages 413–532. Academic Press, 1st edition edition, 2001. ISBN 9780080479569.

[184] G. Szabó and J. Györgyi. Three-dimensional fluid-structure interaction analysis for bridge aeroelasticity. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 2, pages 892–897, 2009.

[185] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers & Fluids*, 1(1):73–100, 1 1973. ISSN 00457930. doi: 10.1016/0045-7930(73)90027-3. URL `https://linkinghub.elsevier.com/retrieve/pii/0045793073900273`.

[186] R. W. Thatcher. Locally mass-conserving TaylorHood elements for two- and three-dimensional flow. *International Journal for Numerical Methods in Fluids*, 11(3):341–353, 8 1990. ISSN 1097-0363. doi: 10.1002/FLD.1650110307. URL `https://onlinelibrary.wiley.com/doi/full/10.1002/fld.1650110307`.

[187] R. W. Thatcher and D. J. Silvester. A locally mass conserving quadratic velocity, linear pressure element. 1 2020. URL `http://arxiv.org/abs/2001.11878`.

[188] D. M. Tidd, R. W. Thatcher, and A. Kaye. The free surface flow of Newtonian and non-Newtonian fluids trapped by surface tension. *International Journal for Numerical Methods in Fluids*, 8(9):1011–1027, 9 1988. ISSN 1097-0363. doi: 10.1002/FLD.1650080904. URL `https://onlinelibrary.wiley.com/doi/full/10.1002/fld.1650080904`.

[189] A. K. Tornberg and M. J. Shelley. Simulating the dynamics and interactions of flexible fibers in Stokes flows. *Journal of Computational Physics*, 196(1):8–40, 5 2004. ISSN 00219991. doi: 10.1016/J.JCP.2003.10.017.

[190] L. R. G. Treloar. The elasticity of a network of long-chain moleculesII. *Trans. Faraday Soc.*, 39(0):241–246, 1 1943. ISSN 0014-7672. doi: 10.1039/TF9433900241. URL `http://xlink.rsc.org/?DOI=TF9433900241`.

[191] S. Turek. *Efficient Solvers for Incompressible Flow Problems*, volume 6 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999. ISBN 978-3-642-63573-1. doi:

10.1007/978-3-642-58393-3. URL http://link.springer.com/10.1007/978-3-642-58393-3.

[192] S. Turek and J. Hron. Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow. *Lecture Notes in Computational Science and Engineering*, 53(June): 371–385, 2006. ISSN 14397358. doi: 10.1007/3-540-34596-5_15.

[193] S. Turek, J. Hron, M. Razzaq, H. Wobker, and M. Schäfer. Numerical Benchmarking of Fluid-Structure Interaction: A Comparison of Different Discretization and Solution Approaches. In *Lecture Notes in Computational Science and Engineering*, volume 73 LNCSE, pages 413–424. 2011. doi: 10.1007/978-3-642-14206-2_15. URL https://link.springer.com/10.1007/978-3-642-14206-2_15.

[194] B. Uekermann, J. C. Carlos, B. Gatzhammer, G. Houzeaux, C. García, M. Mehl, and M. Vázquez. Towards Partitioned Fluid-Structure Interaction on Massively Parallel Systems. In *Proceedings of WCCM XI/ECCM V/ECFD VI*, 2014. URL https://congress.cimne.com/iacm-eccomas2014/admin/files/filePaper/p852.pdf.

[195] K. C. Valanis, R. F. Landel, A. Phys Lett, F. M. Charbonnier, C. J. Bennette, L. W. Swanson, and J. App. The Strain-Energy Function of a Hyperelastic Material in Terms of the Extension Ratios. *Journal of Applied Physics*, 38(7):2997–3002, 6 1967. ISSN 0021-8979. doi: 10.1063/1.1710039. URL /aip/jap/article/38/7/2997/506041/The-Strain-Energy-Function-of-a-Hyperelastic.

[196] A. E. Veldman, J. Gerrits, R. Luppes, J. A. Helder, and J. P. Vreeburg. The numerical simulation of liquid sloshing on board spacecraft. *Journal of Computational Physics*, 224(1):82–99, 5 2007. ISSN 0021-9991. doi: 10.1016/J.JCP.2006.12.020.

[197] H. A. v. d. Vorst. Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *https://doi.org/10.1137/0913035*, 13(2):631–644, 7 2006. ISSN 0196-5204.

doi: 10.1137/0913035. URL https://epubs.siam.org/doi/10.1137/0913035.

[198] J. Wang and A. Layton. Numerical Simulations of Fiber Sedimentation in Navier-Stokes Flows. *Communications in Computational Physics*, 5(1): 61–83, 2009. URL http://www.global-sci.com/.

[199] L. Wang, Z. Liu, and M. Rajamuni. Recent progress of lattice Boltzmann method and its applications in fluid-structure interaction. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 2022(0):095440622210775, 5 2022. ISSN 0954-4062. doi: 10.1177/09544062221077583. URL http://journals.sagepub.com/doi/10.1177/09544062221077583.

[200] Y. Wang, P. K. Jimack, and M. A. Walkley. A one-field monolithic fictitious domain method for fluidstructure interactions. *Computer Methods in Applied Mechanics and Engineering*, 317:1146–1168, 4 2017. ISSN 00457825. doi: 10.1016/j.cma.2017.01.023. URL https://linkinghub.elsevier.com/retrieve/pii/S0045782516309227.

[201] Y. Wang, P. K. Jimack, M. A. Walkley, and O. Pironneau. An energy stable one-field monolithic arbitrary LagrangianEulerian formulation for fluidstructure interaction. *Journal of Fluids and Structures*, 98:103117, 2020. ISSN 10958622. doi: 10.1016/j.jfluidstructs.2020.103117. URL https://doi.org/10.1016/j.jfluidstructs.2020.103117.

[202] A. J. Wathen. Preconditioning and fast solvers for incompressible flow. 2004.

[203] R. Webster. An algebraic multigrid solver for Navier-Stokes problems. *International Journal for Numerical Methods in Fluids*, 18(8):761–780, 4 1994. ISSN 0271-2091. doi: 10.1002/fld.1650180805. URL https://onlinelibrary.wiley.com/doi/10.1002/fld.1650180805.

[204] J. P. Whiteley. A preconditioner for the finite element computation of incompressible, nonlinear elastic deformations. *Computational Mechanics*, 60(4):683–692, 10 2017. ISSN 01787675. doi: 10.1007/S00466-017-1430-3/FIGURES/6. URL https://link.springer.com/article/10.1007/s00466-017-1430-3.

[205] T. Wick. Fully Eulerian fluidstructure interaction for time-dependent problems. *Computer Methods in Applied Mechanics and Engineering*, 255:14–26, 3 2013. ISSN 0045-7825. doi: 10.1016/J.CMA.2012.11.009.

[206] Y. Wu and X.-C. Cai. A fully implicit domain decomposition based ALE framework for three-dimensional fluidstructure interaction with application in blood flow computation. *Journal of Computational Physics*, 258:524–537, 2 2014. ISSN 00219991. doi: 10.1016/j.jcp.2013.10.046. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999113007225.

[207] J. N. Yang, A. K. Agrawal, B. Samali, and J.-C. Wu. Benchmark Problem for Response Control of Wind-Excited Tall Buildings. *Journal of Engineering Mechanics*, 130(4):437–446, 4 2004. ISSN 0733-9399. doi: 10.1061/(ASCE)0733-9399(2004)130:4(437). URL https://ascelibrary.org/doi/10.1061/%28ASCE%290733-9399%282004%29130%3A4%28437%29.

[208] O. H. Yeoh. Characterization of Elastic Properties of Carbon-Black-Filled Rubber Vulcanizates. *Rubber Chemistry and Technology*, 63(5):792–805, 11 1990. ISSN 0035-9475. doi: 10.5254/1.3538289. URL https://meridian.allenpress.com/rct/article/63/5/792/91611/Characterization-of-Elastic-Properties-of-Carbon.

[209] Z. Yu. A DLM/FD method for fluid/flexible-body interactions. *Journal of Computational Physics*, 207(1):1–27, 7 2005. ISSN 00219991. doi: 10.1016/J.JCP.2004.12.026.

[210] C. Zhang. Fluid-structure interaction in rectilinear flows: Four analytical solutions. *Physics of Fluids*, 33(6), 6 2021. ISSN 10897666. doi: 10.1063/5.0055759.

[211] S. Zhang. A new family of stable mixed finite elements for the 3D Stokes equations. *Mathematics of Computation*, 74(250):543–554, 8 2004. ISSN 0025-5718. doi: 10.1090/S0025-5718-04-01711-9. URL `https://www.ams.org/mcom/2005-74-250/S0025-5718-04-01711-9/`.

[212] W. Zhang, Y. Jiang, and Z. Ye. Two Better Loosely Coupled Solution Algorithms of CFD Based Aeroelastic Simulation. *Engineering Applications of Computational Fluid Mechanics*, 1(4):253–262, 2007. ISSN 1994-2060. doi: 10.1080/19942060.2007.11015197. URL `https://www.tandfonline.com/action/journalInformation?journalCode=tcfm20`.