A study of hydroelastic fluid-structure interaction with application to immersed cantilevers

Lorentzon, Johan

2023

Document Version:
Publisher's PDF, also known as Version of record

A study of hydroelastic fluid-structure interaction with application to immersed cantilevers

JOHAN LORENTZON
DIVISION OF COMPUTATIONAL CHEMISTRY | FACULTY OF SCIENCE | LUND UNIVERSITY

Printed by Media-Tryck, Lund 2023 NORDIC SWAN ECOLABEL 3041 0903

LUND UNIVERSITY

Faculty of Science
Division of Computational Chemistry

ISBN 978-91-8096-006-9

9 789180 960069

A study of hydroelastic fluid-structure interaction with application to immersed cantilevers

A study of hydroelastic fluid-structure interaction with application to immersed cantilevers

by Johan Lorentzon

LUND
UNIVERSITY

Doctoral dissertation
Thesis advisors: Prof. Johan Revstedt, Associate Prof. Valera Verayzov, Prof.(em) Per-Erik Austrell, Associate Prof.(em) Per Åke Malmqvist
Faculty opponent: Univ.-Prof. Dr.Ing. Michael Breuer

To be presented, with the permission of the Faculty of Science of Lunds University, for public criticism at Chemical Centre KC:B (floor 1, Naturvetarvägen 14, 223 62 Lund) on Friday, the 1st December 2023 at 9.00

Organization	Document name
LUND UNIVERSITY Division of Computational Chemistry Box 124 SE–221 00 LUND Sweden	**DOCTORAL THESIS**
	Date of disputation 2023-12-01
	Sponsoring organization
Author(s) Johan Lorentzon	

Title and subtitle

A study of hydroelastic fluid-structure interaction with application to immersed cantilevers

Abstract

This study presents an approach for partitioned fluid-structure interaction (FSI) applied to large structural deformations, where an incompressible turbulent solver is combined with a structural solver. The implementation is based upon two different open-source libraries by using MPI as a parallel communication protocol, the packages deal.II and OpenFOAM. FSI is achieved through a strongly-coupled scheme. The solver has been validated against cases with a submerged cantilever in a channel flow to which experiments, numerical calculations and theoretical solutions are available. The verification of the procedure is performed by using a solid-solid interaction (SSI) study. The solver has proven to be robust and has the same parallel efficiency as the fluid and the solid solver stand-alone.

Key words Partitioned Fluid-Structure, Blocked Gauss-Seidel, Jacobian-free Newton-Krylov methods, Incompressible fluid, Finite Strain, Total Lagrangian, Arbitrary Lagrange Euler method, Vector acceleration

Classification system and/or index terms (if any)

Supplementary bibliographical information	Language English
ISSN and key title	ISBN 978-91-8096-006-9 (print) 978-91-8096-007-6 (pdf)
Recipient's notes	Number of pages 127 / Price
	Security classification

A study of hydroelastic fluid-structure interaction with application to immersed cantilevers

by Johan Lorentzon

LUND
UNIVERSITY

Cover illustration front: Collage of cantilever applications subjected to hydroelasticity by the Author on pictures from simulation using Paraview

Cover illustration back: Another collage on the same subject by the Author using the same program as front illustration

Funding information: The thesis work was financially supported by the Swedish Research Council.

Faculty of Science, Division of Computational Chemistry

ISBN: 978-91-8096-006-9 (print)
ISBN: 978-91-8096-007-6 (pdf)

Printed in Sweden by Media-Tryck, Lund University, Lund 2023

Media-Tryck is a Nordic Swan Ecolabel certified provider of printed material. Read more about our environmental work at www.mediatryck.lu.se

MADE IN SWEDEN

My inspiration and greatest passion in my life ... Scientia et Carpe Diem ...

Contents

List of Figures

List of Tables

List of publications

This thesis is based on the following publications, referenced by the Roman numerals:

I **A numerical study of partitioned fluid-structure interaction applied to a cantilever in incompressible turbulent flow.**
J. Lorentzon, J. Revstedt
Int J Numer Methods Eng. 121: 806– 827, 2020

II **On stability and relaxation techniques for partitioned fluid-structure interaction simulations**
J. Lorentzon, J. Revstedt
Engineering Reports. 4(10):e12514. 2022

III **The influence of hydrodynamic damping on the motion of an immersed elastic cantilever**
J. Lorentzon, J. Revstedt, P.E. Austrell
European Journal of Mechanics - B/Fluids, Volume 95, Pages 122-135, 2022

IV **A study of efficiency for partitioned FSI with application to cantilever in transient regime with turbulence modelling**
J. Lorentzon
To be submitted

All papers are reproduced with the permission of their respective publishers. The following contributions to proceedings were pre-study to enclosed papers.

ConfI **A LES study of turbulent flow around twisted and tapered cantilevers**
J. Lorentzon, J. Revstedt
European Conference on Computational Fluid Dynamics—ECCOMAS Barcelona, Spain, July 20-25, 2014

ConfII **Dynamic response and wake structures of cantilevers with square cross-section**
J. Revstedt, J. Lorentzon
MekIT'17, Ninth national conference on Computational Mechanics, Trondheim, Norway, May 11-12, 2017

Other publications not enclosed to this thesis are part of the licentiate "A Theoretical Study of UV Spectroscopy with the CASSCF/CASPT2 Method Applied to Aromatic Systems" 2013 Lund University.

LicI **A CASPT2 study of the valence and lowest Rydberg electronic states of benzene and phenol.**
J. Lorentzon, P.Å. Malmqvist, M. P. Fülscher, B. O. Roos
Theoret. Chim. Acta Volume 91, 91-108, 1995

LicII **A theoretical study of the electronic spectra of pyridine and phosphabenzene.**
J. Lorentzon, M. P. Fülscher, B. O. Roos
Theoret. Chim. Acta Volume 92, 67-81, 1995

LicIII **Theoretical Study of the Electronic Spectra of Uracil and Thymine.**
J. Lorentzon, M. P. Fülscher, B. O. Roos
Journal of the American Chemical Society Volume 117 Issue 36, 9265-9273, 1995

LicIV **Ab initio calculation of inner sphere reorganization energies of inorganic redox couples.**
L. Eberson , R. González-Luque, J. Lorentzon, M. Merchán, B. O. Roos
Journal of the American Chemical Society Volume 115 Issue 7, 2898-2902, 1993

Acknowledgements

I thank my supervisors Per Åke Malmqvist, Valera Verayazov, Johan Revstedt and Per-Erik Austrell for their invaluable guidance of my work. The simulations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at Lunarc and therefore my thanks to the support from the staff/technicians. Then I am grateful to all in the Department of Energy of Science and Theoretical Chemistry. Among those that have been of particular importance for my work, I heartily thanks Cristoffer Norberg, for the guidance in case setup, introducing me to the subject of vortex-induced vibrations, explanation of experiments, and the encouragement to use dimensionless variables and endless resource in references, skip google search, just talk to him.

Nomenclature

Abbreviations

ALE Arbitrary Lagrangian-Eulerian

BC Boundary Condition

BLEND Relaxation technique restricting to the previous iteration only

FCS Forced controlled step

FD Finite Difference

FE Finite Element

FHV Forced harmonic vibrations

FLOPS Floating points operations per seconds.

FV Finite Volume

IB Immersed Boundary

IE Interpolation Error

INS Incompressible Navier-Stokes

JFNK Jacobi Free Newton Krylov

LB Lattice Boltzmann

LES Large Eddy Simulation

MEMS Mechanical electrical micro-structure

PDE Partial Differential Equations

PICARD fixed-point solution without relaxation

Q The Q-criterion, the second invariant of the velocity gradient tensor

RD Rayleight Damping

RTT Reynold transport Theorem

SGS Subgrid-scale

SPH Smoothed-particle hydrodynamics

Symbols

Γ	The coupled interface between fluid and solid
ν	Fluid kinematic viscosity
ν_s	Poisson ratio
σ	The stress
T	Second Picola-Kirchhoff Stress tensor in matrix form
τ	The shear stress
ζ	Damping ratio
B_L	Linear second order strain-displacement in matrix form
B_{NL}	Nonlinear second order strain-displacement in matrix form
D	The width of the cantilever
d	The thickness of the cantilever
E	Young modulus
f	The frequency of the structure immersed in the fluid
f_v	The frequency of the structure in vacuo
L	The height of the cantilever
p	The pressure
R_d	Gain factor, the ratio of dynamic field value to the quasi static value
t	traction
U	Magnitude of inlet velocity for channel flow
R	Mass ratio of density of solid to fluid

Notation

The reader should be aware of notation overlap between disciplines, unless presented in the abbreviation list, the given notation applies in the given context.

Einstein notation

Einstein notation is only uses the latin letters between i and m, and contraction applies to tensor/matrix terms over subscript to subscript or subscript to superscript. No separation (comma/punctuation) is added to equations displayed as multiline or ending with a variable using a comma in the subscript.

- $a_k b_k = \sum_k a_k b_k$
- $a_k b_i^k = \sum_k a_k b_i^k$
- $a_k 1_k = \sum_k a_k$
- $a_{i,j} = \frac{\partial a_i}{\partial x_j}$

Pseudo function

Pseudo functions, often expressed in calligraphic font or while aggregating tensor expression into matrices, are with omitted indices

- $\mathcal{F}(a_i) \Leftrightarrow F : a_i \rightarrow b_i, a_i \in X, b_i \in Y$
- $B_{NL} T B_{NL} \Leftrightarrow (B_{NL})_{ik} T_{kl} (B_{NL})_{lj}$

This also applies to the mapping functions in finite strain theory and ALE procedure.

Voigt notation

The Voigt notation allows conversion of second order symmetric tensor to be expressed as vectors, for example Caucy stress tensorm σ_{ij} $[\sigma_{xx}, \sigma_{rr}, \sigma_{zz}, \sigma_{xt}, \sigma_{zx}, \sigma_{yz}]$. In order to retain expression of governing equations involving partial derivatives, replace the derivatives of tensor with an operator matrix such that, for example, $\sigma_{ij,j} = L_{ij}\sigma_j$.

Populärvetenskaplig sammanfattning på svenska

Inom beräkningskemi arbetar man mer med molekyler än atomer, studerar stökiometriska relationer, kinetik och geometri, interatomära eletronövergångar medan beräkningsfysik studerar mer atomer och dess kärnor, fasta tillståndet och intraatomära övergångar, oftast närmast mot atomens kärna. Interaktionerna sker på en storleksskala på nanometer eller mindre med energiövergångar som är diskontinuerliga, kvantiserade tillstånd, vilket utgör en av kvantmekanikens hörnstenar och beskriver båda disciplinerna. Med ökat antal atomer börjar interaktionerna mellan atomerna sudda ut identiteten av de individuella bidragen, och den fysikaliska beskrivningen övergår till statistiska beskrivning av ensemble av atomer/molekyler, gradvist övergående till medium av materia, tillstånd med dess lägesenergier och hastighet av materie punkter, till den grad att dessa ändringar blir synliga för ögat och den statistiska partikel beskrivningen övergår till tillstånds variabler, den klassiska mekaniken med Newtons lagar i dess centrum med kontinuerlig variation av energi. Dessa tre discipliner, utgör de grundstenar till vilket ingenjörskunskap vilar på, vars konstruktioner utgör grunden för vår moderna civilisation. Konstruktioner involverar oftast växelverkan mellan olika material tillstånd. De fundamentala material tillstånd man oftast talar om är gas, flytande och fasta tillståndet och energiövergångar är material punkternas ändring i kinetik, den inre energin med dess övergångar uttryckt i ändring av antingen tryck, värme, lägesenergi, elektricitet eller magnetism. I studie av flöde och kraftpåverkan under normala förhållande, som påverkar vardagen och dess mekaniska tillämpningar är mest centrerad kring gravitation, tryck och kinetik av kroppar. När man åtsidosätter energetiska dissipationer och permanenta form ändringar av kroppar så studerar man elastiska interationer. När man dessutom begränsar sig mot interaktionen mellan fasta strukturer, dvs konstruktioner, och vätskor, hamnar man inom ämnet fluid-struktur interaktion (FSI) med hydroelastisk tillämpning. Mer specifikt rör det sig om vibrationer, töjningar, sträckning av strukturer, som mäts iform av frekvens och utböjningsstorlek. Detta omfattar ett område som har rönt stort intresse under de senaste årtiodena, från makroskopisk skala med marina tillämpningar som oljeplattformar, rörsystem under vatten och transport av vätskor i processindustri, ned till mikroapplikationer som t.ex. sensorsystem för mätning av tryck och flöde. Att minimera utmattningsskador/slitage som uppstår genom strömning som inducerar vibrationer i struktur är ett väsentlig moment i design, konstruktion och underhåll. I många applikationer strävar man även efter att maximera interaktionen, tex vinge/rotor/turbin och förbränningsmotorer. Att uppskatta effekten av FSI gör man genom experiment där man ofta skalar ned applikationen till modellstorlek, och sedan genom vindtunnlar/bassänger studerar kraftöverföring och mäter frekvens och amplituder på inducerade vibrationer. Sådana experiment är ibland inte genomförbara, och därför är oftast mätning på applikation i dess naturliga storlek det enda sättet för att förut-

säga dessa effekter, experiment eller simulering. Detta arbete går igenom metodiken kring hur man skapar simuleringsverktyg, undersöker dess parameterberoende och utvärderar dess tillämpbarhet. Applikationsområdet i denna studie är gummipelare nedsänkta i vatten i meterskala, där flöde induceras av rörelse av strukturen, eller låter man kanalströmning generera rörelse av strukturen, varmed FSI skapas. Detta är geometriskt och kinematiskt ekvivalent genom likhetstransformation med sensorsystem i mikroskala där kiselstrukturer nedsänkta i vätska mäter t.ex. tryck/flödeshastighet. Ett program som löser FSI problemet i helhet, både fluid och struktur samtidigt, kallas monolitlösare, men man kan även lösa detta genom fixpunktslösning, där in och utdata från fristående program på struktur och fluid överförs mellan varandra genom ett kopplingsschema tills konvergens av data uppnås. Att kombinera två existerande program på detta sätt kan vara det enda praktiska sättet att skapa en metodik som löser FSI problem, eftersom det oftast kräver omfattande mansarbete bakom att skapa en monolitlösare och/eller tillgång till öppen källkod är begränsad. Detta tillvägagångssätt brukar betecknas som partitioneringsmetodik och präglas av instabilitet och oftast begränsad till aerodynamiska applikationer. Denna studie presenterar en partitionerad FSI lösare baserad på två avgränsade struktur och fluid källkod, deal.II respektive OpenFOAM. Lösaren stabiliseras genom relaxationstekniker och kopplingschema. Som resultat av detta har en lösningsmetodik utvecklats som är stabil och effektiv för hydroelastiska applikationer som kan tillämpas på ingenjörsmässiga applikationer men även inom grundforskning kring strömnings fenomen skapat av FSI, tex resonans och vakmekanismer.

Preface

Assuming the reader of this thesis is a quantum chemist who typically computes the kinetics and reaction mechanics and is working with his code, a state solver for quantum mechanical modelling, how could a solution procedure for a state solver in macrostate be of any relevance in his line of work? Or vice versa, a computational engineer working with the implementation of solution procedure of fluid-structure interaction (FSI), why should he pay any attention to the work of a computational chemist?

The answer to this, despite the difference in the application area, is that both may require the use of the same platform, especially while dealing with highly costly computations, High-Performance Computer Centre (HPCC) and thus requiring parallelisation. Further, the methodologies to solve state equations, namely, solving fixed-point problems and applying relaxation techniques, are mostly the same. This chapter works through the motivation for changing the subject, the connecting dots between the microstate and macrostate, and the layout of the thesis. This preface is a caveat for presenting the subject of FSI in the discipline of computational chemistry using an application of a cantilever immersed in channel flow, the multiscale ladder picture.

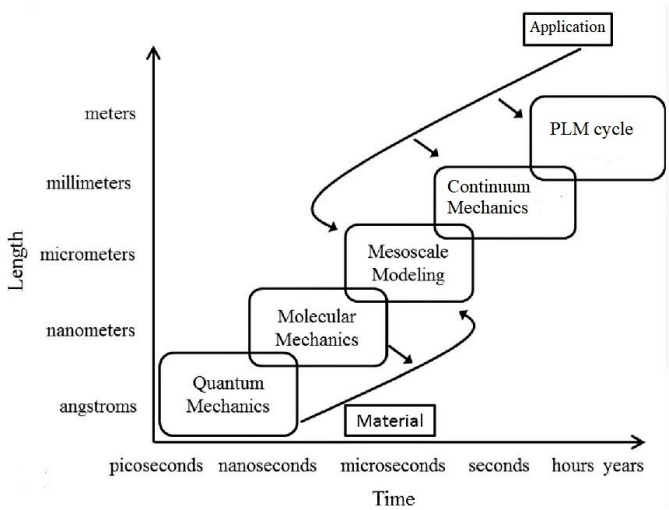

Figure 1: Multiscale ladder. The handshake between the Licentiate and PhD thesis. The PLM is the Production Life Management cycle: a design, construction, and maintenance process cycle in which vibrational analysis for fatigue and stress cycles plays a central role in the estimation of the lifespan of an application.[Google search: Corpus ID: 133876590 (reworked)]

Previous work as part of the thesis work

By tradition, the midterm of the PhD candidature is marked with writing a Licentiate, which was on the subject of computational chemistry. For the remaining part of the PhD, the work was shifted towards computational multiphysics. The articles covered in this licentiate concern electronic spectra of aromatic molecules and atmospheric chemistry. With computational chemistry, one, for example, could compute reaction kinetics, which can be used in combustion chemistry, to which continuum theory can be applied to compute flame propagation, see for example Lic IV. The solving of Schrödinger equation often comes down to fixed-point formulation, using relaxation/stabilisation techniques. Here the perturbation methodology and relaxation techniques, applied in Lic I to Lic III has been most useful in the approach to stabilise the FSI procedure. The chosen application, the cantilever, can be applied from macrostate to microstate, and while used together, it is often phrased as a multiscale ladder approach; see Figure 1. It should be noted, however not the topic in this work, that material description, for properly handling fatigue, may require modelling at the mesoscopic range. The incitement for changing the subject was twofold: firstly, the author's interest in parallelisation, and secondly, the authors's interest in solving partial differential equations (PDE) by different methodologies, finite element (FE), and finite volume (FV) methods.

The layout of the thesis

It will begin with the introduction of the problem, the scientific goals, and the motivation of the subject. Then the chapter on methodology covers FSI boundary conditions and the governing equations, the applied partition techniques, and the stabilisation. The chapter on methodology thus defines the scope and the terminology/abbreviations used for the coming chapters. To set the work in context with other research, an overview of the FSI is given, the literature study, the background on the subject, and what challenges are expected, with a focus on the application, the cantilever. Then the solution procedure for solving the FSI is presented, followed by a chapter on the implementation and validation of the FSI solver. The reason for separating these two is that implementing a procedure often has sharing concepts with different solution procedures. With the FSI solver verified, and methodology explained, a chapter is dedicated for the summary of the results from the enclosed articles together with connecting the dots between the articles, focusing on the research goals of interest. An outlook for future and ongoing work for the papers that are in progress but yet not to be presented here.

Chapter 1

Introduction

Fluid-structure interaction (FSI) is a multidisciplinary field that covers a large field of engineering applications, ranging in size from macroscopic down to microscopic. The FSI induces vibrations and fatigue to the structure, and the redirected flow around the structure generates a flow pattern that may destabilise the structure by coherency over a large distance. Designing applications subjected to FSI requires quantifying these effects, and experimental input may be limited. Simulation tools can aid the engineer in pursuing this agenda. However, software development and computer architecture requirements to achieve this are costly, and continuous research is ongoing to increase efficiency and application range. This introduction covers the background of the FSI subject, the keywords used in later sections to place it into context, and a description of the application. The problem description of the FSI and the goals are also formulated.

1.1 Background and scope

The redirected flow around a structure may generate a beating with a specific frequency that changes linearly with the incident flow speed and causes deformation of the structure normal to the flow direction. Assuming starting from low flow speed and increasing, an immersed structure with the flow may respond with a resonance motion with a locked-in structural frequency or galloping/fluttering with increasing amplitude with no lock-in frequency response. The difference between galloping/fluttering is mainly the choice of application. Both types begin at the onset of a critical flow velocity. Some typical applications are wings, blades, and immersed cables/frameworks. The resonance often occurs in regimes of reduced flow speed that generate a beating near the immersed structural frequency, characterised by the width and mass ratio between fluid and structure. The frequency of oscillatory motion and the amplitude define the fatigue cycles to which the material stability and endurance relate. In addition, a turbulent flow induces fatigue stress commonly seen in turbines/hulls caused by buffeting/cavitation. A complexity with FSI is the added mass effect of fluid moving coherently with the structure, which induces shift in the frequency of the structures, usually lowering it. Also, the structure's damping normally increases but may also reduce even to negative damping, typical for galloping/fluttering. The influence of FSI can thus be measured by the structural deformation relative to an equivalent force magnitude in quasi-static (gain factor), the damping and the frequency shift.

By design, one seeks to minimise/maximise the gain factor by either passive or active FSI, depending on the application. Passive FSI is often achieved by optimising the shape of e.g. a ship hull, an aeroplane body, or a blade of a turbine/propeller/wing, or by modifying the flow pattern of submerged pipelines by twisting or tapering, or counteracting triggered mode shapes in vessels, bridges and buildings by reinforcement over nodal points of higher order vibrational modes. Adaptively changing the gain factor of structure, the active FSI, often refers to dynamically altering the stiffness/shape, such as blade twisting in turbine/helicopter blades and actuators/dampers in bridges/buildings. In pursuing shape optimisation, one often resorts to cyclic iterative procedures by experiments, where the state variable's response is measured by varying structural parameters one at a time. From these inputs, one forms a functional expression of the dependency between structural and flow parameters. These empirical expressions often rely upon dimensional similarity analysis, providing the engineer with diagrams and formulas to be used in the design. For example, statistics of mechanical failure are related to fatigue, which relates to the size of the induced motion and its frequency, or a diagram of hull shape for a ship/aeroplane as a function of load capacity/cruiser speed. However, experimental input may not be feasible or not provide sufficient data for various reasons; therefore, simulation tools are the only viable option. FSI methodology specially adapted to these challenges has

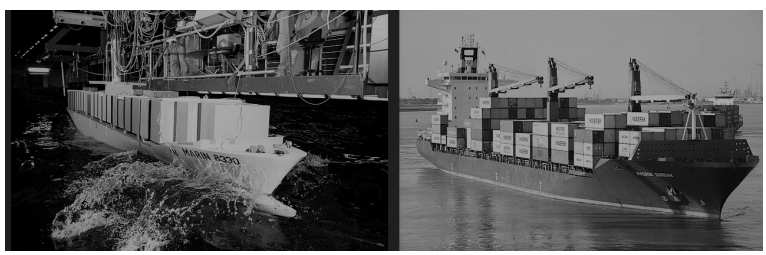

Figure 1.1: Left: Modelling tanker under rough sea condition in a basin. Right: The corresponding ship type in voyage with cargo. [Wikimedia Commons search: maersk basin]

been developed, such as immersed boundary method for simulation of a heart valve[4], space-time discretisation[5] with unique adaption for the description of a parachute, a smooth particle model for astrophysical problems[6], and Lattice Boltzmann to handle the complex transition between phase transition/complex media such as porous media or droplets[7]. Other examples are the volume of fluid method for handling multiphase flow, such as in the mixing of fluids and sloshing tanks[8], the added mass methodology for steady-state analysis of stress over hulls of ships/storage tanks[9], and the Method of characteristics to analyse water hammer in pipework[10]. A similar approach, the Morison equation[11], has been developed for wave interaction with immersed frameworks in ocean applications. All these methods have the common ground of solving FSI within continuum theory and elasticity.

Despite having simulation tools that handle specific design problems, implementing them for each program is demanding; therefore, having an FSI methodology that allows handling each program as a black-box is an attractive feature, where the state variables are provided in a cyclic fashion as alternate input/output. However, such a partitioned approach suffers from severe limitations from instability, especially for hydroelastic problems (i.e. FSI where the fluid is liquid); this is because the mass ratio between the fluid and the structure affects the load that is transferred with respect to momentum; the larger the momentum, the larger excitation of the cyclic loading, leading to larger fatigue. Hydroelasticity often occurs in offshore applications, such as oil-rig platforms, ships, bridges, submerged pipelines, pipeworks transporting liquids, and storage facilities of liquids in containers. Here one distinguishes between quiescence flow and channel flow. A quiescence flow often occurs in servo-pumps, storage tanks under mechanical stress or sensor arrays at the microscale (MEMS). Channel flow is the transport of fluid, often pipes.

This thesis focuses on developing a partitioned FSI methodology that is applicable to turbulent flow with hydroelasticity using a cantilever immersed in quiescence and channel flow as the primary technical application. Challenges that come with such partitioned methodology are investigated and characterised.

1.2 The application

In most engineering structures, there is an interaction between fluid and solid matter to be considered. To analyse FSI effects, one often begins with identifying critical components in the structure subjected to FSI. One such is the cantilever, which acts as a support element in engineering frameworks, from moving mechanical machinery on land, in/on water, or in the air, or likewise stationary applications, such as oil platforms, large-scale buildings, plants with large pipe structures, and so forth. Viewing a landscape of a city or industrial plant, most structures are cantilever-supported structures, a reminder of how this component has such widespread usage. In the opposite size range, at the microscopic level, mechanical devices measure temperature, flow speed, or pressure, or act as transducers, actuators, gears, pumps, and switches. Another aspect of the cantilever/beam is its simplicity in providing analytical expression to model more complex structures, mechanical ones such as ships (free beam) and wings/buildings/tails (cantilever), biological structures such as grass, seaweed or trees (cantilever), and so forth. The application is a cantilever immersed in a rectangular domain of fluid which is either in motion (channel flow) or still tank (quiescence flow). This can be related to numerous engineering problems, as exemplified above, and thus serves as a suitable ground for a thorough investigation of FSI methodology. Also, with cantilever, a well-defined theory upon FSI and beam theory is another

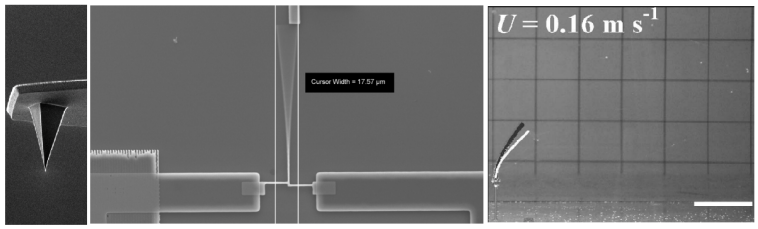

Figure 1.2: Cantilever applications from nanoscale, left most, AFM cantilever, most used for taking pictures of material structure, MEMS resonance for the purpose of measuring, up to macroscale - laboratory scale (cm), studying deflection of thin beam foam immersed in channel flow. By similitude, results at the macroscale can be brought to be similar to case settings at the microscale. [Wikimedia Commons search: AFM, MEMS] and rightmost Luhar [1]]

attractive point in the chosen application. Choosing a rectangular cantilever cross-section creates the point of separation of shear layer located at the edges, making the immersed cantilever in a fluid domain cost-efficient for extensive parameter studies.

A hyperelastic model of the solid is chosen, since the aim is for medium-sized deformations, with a material Lamé parameter setting similar to rubber. As turbulence is an important aspect, the focus is set on the transition regime, moderate turbulence mixed with the laminar flow by further setting uniform/parabolic laminar flow settings, leading to turbulence generated by the flow separation around the bluff-body, which further scales down the computational efforts. The focus on the hydroelastic problems is provided by setting the density and viscosity of the fluid similar to water. The domain is in meter scale; however, by similitude, the chosen application and settings are equivalent to silicon/water for micro-scaled applications (μm), such as sensor systems, or laboratory scale (cm) using glycerine/water mixtures.

Consequently, the Mach number is low, this makes incompressible fluid condition the most sensible choice since the stability of the solver decreases as the Mach number decreases for a compressible medium. On the contrary, an incompressible solid would cause problems for its solver, making a compressible condition on the solid the most suitable for the current material setting.

1.3 Problem description of FSI

In the design of an application/control/investigation of a physical process, one has to choose the governing variables for a given physical state. In this choice of variables, conditions, and kinematics follow since implicit dependencies apply as a consequence of physical laws.

Most physical laws revolve around mass, momentum and energy conservation. State

variables often govern a physical state, for example, pressure, velocity and mass, but also by electrical or thermal origin conditions. Together it generates a set of interrelations between state variables, the governing equations, which may require further modelling, to create the equation of state solver whose solution provides the governing state variables. Solving the governing equation of state for a multi-physic problem in three dimensions often requires computer-aided resources such as high-performance computers (HPC) clustered into massively parallel systems.

Today many open-sourced or commercial solvers solve fluid and solid problems stand-alone in an HPC environment in an efficient and scalable way, but there are much fewer solvers that combine those different disciplines to the same degree of success. A common approach to solve an FSI problem is to exchange state variables (x) across the coupled boundary through an interface and formulate the governing equations of state as a fixed point problem, where each sub-domain is solved sequentially by its stand-alone solver,

$$x = \mathcal{H}(x) \equiv \mathcal{F} \circ \mathcal{S}(x), \tag{1.1}$$

where the solvers $\mathcal{F}$ and $\mathcal{S}$ is applied to the fluid respectively solid sub-domain. This partitioned black-box procedure generates a sequence $\mathcal{X} = \{x_i\}$ whose limit x^* is the desired solution (x); however, this procedure may be unstable, mostly due to the so-called added mass problem that appears in hydroelasticity applications. An improvement of this procedure may be obtained by the application of vector acceleration techniques to the sequence, where the sequence $\mathcal{X}$ is transformed by mapping each element by a function f,

$$f : \mathcal{X} \mapsto \mathcal{X}' \equiv \{f(x_i)\}. \tag{1.2}$$

The choice of the function f depends on the design parameters, and since FSI for hydroelasticity problems in engineering applications tends to be stiff, an efficient solution is often difficult to achieve. This is further complicated because most large-scale flows are turbulent, providing noise which may cause the accelerating procedure to fail. Another approach is to apply the extrapolation/projection technique; the difference lies in the reformulation of the problem into the residual,

$$\mathcal{R}(x) = \mathcal{H}(x) - x, \tag{1.3}$$

then by applying a root-finding procedure to solve the problem. The functional representation of state variables is normally achieved using a mesh, a tesselation of the domain into sufficient small subdomains such that points in space can be related to each other by compact functional expressions, thus providing a discrete representation of state variables that varies across the space. A change of the boundary implies motion of the mesh; this poses an additional complexity that is either solved by a superimposed stationary mesh, immersed boundary technique (IB), or coherently moving disjoint

mesh, normally solved by the Arbitrary-Lagrangian-Eulerian technique (ALE), thus enforcing a set of Laws of Geometric constraints that has to be enforced into the balance equations for the fluid/solid. This is employed normally as part of the fluid step but can be treated as a stand-alone solver $\mathcal{M}$ with the input of the boundary, and the output is the change in the fluid mesh.

1.4 Research goals and objective of thesis

The main objectives of this thesis were to study the influence of turbulence and evaluate the feasibility of partitioned FSI methodology by choosing a cantilever and applying this to an area that at the start of the project was not thoroughly investigated, namely, FSI in turbulent flow for hydroelastic problems by setting the material close to rubber/water. The solution approach is a partitioned technique using ALE methodology and relaxation to enforce stability.

Although the main goals were achieved in the first paper, which was a significant novelty when published, the turbulence showed only a minor influence with the current application and required significant computer resources to further investigate the limitation while using ALE methodology with turbulence. Therefore, the focus shifted from turbulence modelling to studying the efficiency and stability of relaxation techniques.

By choosing cost-efficient settings for the application, the stability and efficiency of the FSI methodology were thoroughly investigated. This involves parameter dependency studies on stand-alone solvers, coupled solvers by different schemes and domain studies. The primary goal was to establish a procedure for a stable and efficient FSI simulation.

Also, while shifting from turbulence modelling into hydroelasticity/hydrodynamic damping, a secondary objective was to investigate vortex-induced vibrations (VIV) to demonstrate the capability of the FSI methodology in an area with large ongoing research. Since the application often requires damping in the solid, modelling aspects of the damping becomes an additional goal.

Chapter 2

The methodology

The chapter first describes the governing equations, the boundary conditions, and the modelling equations for turbulence. Then focus is on the FSI methodology, the partitioned approach, the state diagram representation of the solver, and then the relaxation methodology defined. See Appendix A for basic terminology and dimensionless numbers.

2.1 The governing equations

The physical domain Ω is partitioned into two disjoint subdomains, the solid Ω^s and the fluid Ω^f, with a coupled boundary Γ. Each state variable is a function of position x_i, and for each physical or fictitious boundary, a normal vector (n_j) is defined and governed by a balance equation in strong form. The state variable assigned to each domain has superscript s or f, which is skipped whenever the context is clear or a relation is the same for both domains. At Γ the Dirichlet-Neumann condition is applied,

$$v_i = a_{i,t} \tag{2.1}$$

$$t_i^s = -t_i^f \tag{2.2}$$

where v_i is the boundary velocity on the fluid side and displacement vector on the solid side and t_i is the traction. This is a continuum hypothesis (i.e. no-slip) and Newton's third law for action and reaction, respectively. Assuming a material element with control volume CV of arbitrary size, and studying the change of a quantity ϕ over time integrated over the CV, the Reynolds transport theorem (RTT) states the relation between the flux across the surface with velocity v_k and the change within the CV to the measured total change of the quantity expressed as,

$$\frac{d}{dt}\int_{CV}\phi dV = \int_{CV}\phi_{,t}dV + \int_{\partial CV} n_k v_k \phi dA. \tag{2.3}$$

This is the Lagrangian view. alternatively, by changing v_k to u_k, assuming a static fixed CV and studying the flow of the fluid, the Eulerian view. Normally it is Lagrangian view applied to the solid and the Eulerian view to fluid. Further, normally for solids, one set v_k=0. Applying RTT to formulate the balance equation for the mass, $\phi = \rho$, in each subdomain gives the continuity equation,

$$\rho_{,t} + (\rho U_i)_{,i} = 0, \tag{2.4}$$

where ρ is the density and U_i the velocity. For incompressible flow, ρ being constant and Eqn (2.4) is simplified to $U_{i,i} = 0$. Correspondingly, for the Lagrangian description of a solid, Eqn (2.4) has a closed algebraic form, $\rho_0 = J\rho$ where subscript 0 refers to density in the reference frame, and J is the determinant of the deformation gradient of the strain measure. For fluid domain, using RTT to formulate the balance equations for momentum, with $\phi = \rho U_i$, incompressibility ($U_{i,i} = 0$), Newtonian fluid ($\tau_{ij} = 2\nu D_{ij}$), Newton's second law ($\rho\frac{dU_i}{dt} = f_i$), Cauchy stress theorem, with Cauchy stress formulation, thus expressing the force in terms of shear (τ_{ij}) and pressure(p), and divergence theorem gives the incompressible Navier-Stokes equations (INS),

$$U_{i,t} + U_j U_{i,j} - 2\nu D_{ij,j} = -\frac{1}{\rho} p_{,i} \tag{2.5}$$

where ν is the kinematic viscosity. There are four variables (U,x,p,t), and commonly one applies the following scaling ($U^\dagger = \frac{u}{U}$, $x^\dagger = \frac{x}{L}$, $p^\dagger = \frac{p}{\rho U^2}$, $t^\dagger = \frac{t}{L/U}$) where L is the so-called length scale, U reference magnitude, thus expressing the INS in terms of the Reynolds number (Re),

$$U^\dagger_{i,t^\dagger} + U^\dagger_j U^\dagger_{i,j} - 2\frac{1}{Re} D^\dagger_{ij,j} = -p^\dagger_{,i} \tag{2.6}$$

The choice of scaling reflects the type of problem to be solved, and Reynolds number characterises, among other things, the turbulence in a flow.

The corresponding momentum equation for solids takes, due to the motion description, a much more compact form, the equation of motion, which relates the volume forces with the acceleration and density,

$$\rho a_{i,tt} + f(a_i, a_{i,t}) = f^e_i, \tag{2.7}$$

where f is the internal force, f^e is the external force, a_i the displacement field. The balance equations to be solved are coupled through the common boundary Γ. Commonly, $f(a_i, a_{i,t}) = h(a_{i,t}) + g(a_i)$ and applying linear damping ($h \equiv \eta a_{i,t}$) and Cauchy stress theorem ($g \equiv -\sigma_{ij,j}$), Eqn (2.7) takes the following form,

$$\rho a_{i,tt} + \eta \dot{a}_i - \sigma_{ij,j} = f^e_i. \tag{2.8}$$

The BC at the coupled boundary, governed by Eqn (2.2), implies that the traction by the fluid acting on the solid, scales by the mass ratio $R = \frac{\rho_f}{\rho_s}$.

2.2 Turbulence modelling

The turbulent flow contains a cascade of degrading motion that is of a chaotic nature with a self-similarity, thus requiring finer mesh to be resolved. The exact velocity field

U_i (unknown) is therefore decomposed into a resolved field (i.e. known field) $\bar{U}_i$ and a subgrid-scale part u'_i (modelling portion of the cascading part of the turbulence),

$$U_i = \bar{U}_i + u'_i, \tag{2.9}$$

where the resolved field is $\bar{U}_i = \int G(r)U_i(x - x', t - t')dx'dt'$, of with a filter function G. Applied to Eqn (2.5), this defines the sub-grid part in terms of the resolved component, and the filtered INS in strong form is obtained[12],

$$\bar{U}_{i,t} + \bar{U}_j\bar{U}_{i,j} + B_{ij,j} - 2\nu\bar{D}_{ij,j} = -\frac{1}{\rho}p_{,i} \tag{2.10}$$

where B_{ij} is the sub-grid scale (SGS) stress tensor, $B_{ij} = \overline{U_iU_j} - \overline{U}_j\overline{U}_i$. The applied turbulence model is the Large Eddy Simulation model (LES).

Two main LES approaches exist, explicit[13] and implicit filtering[14]. In the explicit form, a filter is applied, and the SGS turbulence is computed explicitly, while in the implicit, a top-hat filter and dispersion of the discretisation scheme is used, the upwind scheme for example.

A common approach to finding the closure to the Navier-Stokes equation, namely parameterise B_{ij}, is to assume that the turbulence is aligned with the strain tensor of the mean flow, the Boussinesq hypothesis[15]. Although this is often not the case, it has one aspect that makes it still useful: even a non-exact turbulence model at least has the advantage that it mitigates the stability loss that would result when trying to solve INS without turbulence for large Reynolds number. However, one known issue with LES is the non-commutative property[16] in moving parts, introducing an error. The thesis does not cover this topic, but its effect is limited by the application.

2.3 Solving the balance equation

The tensor fields are continuous functions over the whole space. Applying spectral methods, using a continuous basis function over the entire domain, is feasible but not a common practice for various reasons. To achieve a solution to the balance equations, one needs to simplify the balance equations into a finite representation by discretisation; this is achieved by first defining the mesh of a domain. The domain is partitioned into a disjoint set of control volumes (CV), polyhedrons, defined by vertices which form facets, and each facet has an associated normal. This forms the mesh of the domain, where the state variables take discrete values indexed by either the vertices or the centroid of the control volume or the facet. The localisation matrix Z_e relates the local elements X_i^e with the global field X_i.

$$X_i^e = (Z_e)_{ij}X_j, \tag{2.11}$$

The balance equations can then be solved "exactly" for each control volume with a local balance equation $\mathcal{A}^e(X_i^e)$ for a non-linear root problem. The assembling of these interrelated equations for each control volume allows the strong form of the governing balance equations to be formulated as the aforementioned fixed-point problem,

$$X_i = \sum_e \mathcal{A}^e(X_i^e) = \sum_e \mathcal{A}^e((Z_e)_{ij} X_j) \equiv \mathcal{A}(X_i), \tag{2.12}$$

where $\mathcal{A}$ is the assembled equation, often expressed as recurrence relation or functional expression. The process of assembly and formulating this is phrased as the solution procedure with $\mathcal{A}$ as the solver of the balance equations. The classification of the methodology is independent of the choice of the solution procedure. By designating $\mathcal{A}$ to $\mathcal{F}$ for the balance equations of the fluid domain and $\mathcal{S}$ over the solid domain, one obtains a solver for each subdomain. The unknown variables of displacement and traction across the coupled boundary are thus a_i respectively t_i. By defining the mapping $a_i = \mathcal{S}(t_i)$ and $t_i = \mathcal{F}(a_i)$ at the coupled boundary, two different solution approaches are,

$$a_i = \mathcal{S} \circ \mathcal{F}(a_i) \vee \ t_i = \mathcal{F} \circ \mathcal{S}(t_i), \tag{2.13}$$

and

$$\begin{bmatrix} a_i \\ t_i \end{bmatrix} = \begin{bmatrix} \mathcal{S}(t_i) \\ \mathcal{F}(a_i) \end{bmatrix}, \tag{2.14}$$

respectively. One could solve this simultaneously in the same mathematical framework, monolithic solution, or sequentially in each subdomain. While solving this for a monolithic approach, the Eqn (2.14) is the most natural choice, the Jacobi procedure, however, by solving each domain sequentially using the coupling, Eqn (2.13) is a natural choice, using the Gauss-Seidel procedure. The composite $\mathcal{S} \circ \mathcal{F}(a_i)$ used in formulating the FSI problem is designated as $\mathcal{H}$, a cornerstone in the FSI methodology. Whenever the framework is separate for each subdomain, hence ignoring the coupling, the sequential procedure is also known as the partitioned approach.

2.4 A mesh conforming motion technique

The ALE method[2] is a technique where one solves the fluid equation by moving the whole grid on the fluid subdomain in a Lagrangian perspective, where the fluid solver remains in Eulerian perspective, originally formulated for moving boundary problems without the structural subdomain[17]. The Lagrangian perspective solves the balance equation in the material domain and is related to the spatial domain by a mapping φ, see Figure 2.1. This mapping φ is equivalent to $\Phi \circ \Psi^{-1}$ which relates the gradients in

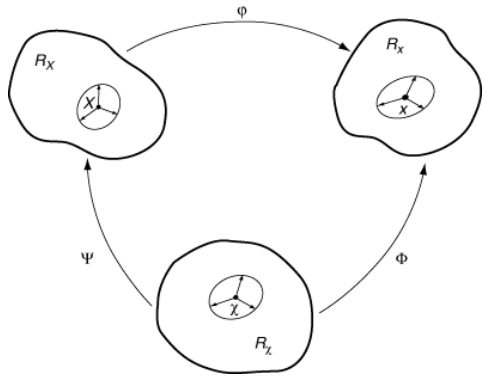

Figure 2.1: ALE mapping: The reference domain R_χ maps the position of the material space and spatial space by the mapping Ψ respectively Φ which produces the mapping (φ) between material point X_i to spatial point x_i through the composite mapping $\phi \equiv \Phi \circ \Psi^{-1}$ [Chapter 14: Arbitrary Lagrangian-Eulerian by Donea [2]].

the material and spatial domain, (X_i,t) respectively (x_i,t), thus following the motion of a material point in the spatial domain and differentiate the ϕ with respect to X_i to find the relation between the velocities in each domain by chain rule,

$$\begin{bmatrix} \frac{\partial x_i}{\partial X_j} & v_i \\ 0_i^t & 1 \end{bmatrix} = \begin{bmatrix} \frac{\partial x_i}{\partial \chi} & \hat{v}_i \\ 0_i^t & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial \chi}{\partial X_j} & w_i \\ 0_i^t & 1 \end{bmatrix} \equiv \begin{bmatrix} \frac{\partial x_i}{\partial X_j} & \hat{v}_i + \frac{\partial x_i}{\partial \chi} \cdot w_i \\ 0_i^t & 1 \end{bmatrix}, \tag{2.15}$$

where $(v_i, \hat{v}_i, w_i) \equiv (\left.\frac{\partial x_i}{\partial t}\right|_x, \left.\frac{\partial x_i}{\partial t}\right|_\chi, \left.\frac{\partial \chi}{\partial t}\right|_X)$. This is used to relate the derivatives of a variable f in spatial with respect to the material through the reference, ($f(X_i, t) = f(\chi, t) \circ \Phi^{-1}(x_i, t)$) by using the chain rule,

$$\left.\frac{\partial f}{\partial t}\right|_X = \left.\frac{\partial f}{\partial t}\right|_\chi + (v_k - \hat{v}_k) f_{,k} \tag{2.16}$$

where $(v_i - \hat{v}_i)$ is the relative velocity in reference space and $\hat{v}$ is the mesh velocity.

As a consequence of this, the governing equations for the fluid, which are formulated in the Eulerian description, can be related to the Lagrangian description by a mere change of velocity in the Eqn (2.5) for the convective part ($U_k U_{i,k}$) with relative velocity (i.e. $U_k^r U_{i,k}$). Computing the mesh motion velocity U_i^r requires an additional solution step, the mesh motion solver, which is denoted as $\mathcal{M}$ with the structural interface coordinates as input, and the output is the mesh velocities of the fluid domain, i.e. $v_i = \mathcal{M}(a_i)$.

Also, due to the motion of cells, to preserve the conservation laws (i.e. balance equations), the Geometric Law of Constraints (GLC) [18] must be applied unless one deals with space-time formulation. A commonly applied solution approach is to enforce

uniform flow preservation, which implies a correction step by adding a source term that removes the error produced by solving an advection problem for constant flow,

$$\left.\frac{\partial}{\partial t}\right|_{\chi} \int_{CV} dV = \int_{\partial CV} u_i n_i dS. \tag{2.17}$$

This is a procedure that has been debated over decades, notably for FV methods, about its necessity and sufficiency, but in general it has been considered to be a key ingredient for increased stability and accuracy [19,20,21]. The mesh motion procedure is closely dependent on the meshing strategy. The applied mesh motion procedure in this thesis is the diffusion algorithm [22,23], which assumes the mesh topology is fixed (i.e. same interrelations and cells),

$$(\gamma d^f_{i,jj}) = 0 \quad , \quad v^g_i = d^f_{i,t} \tag{2.18}$$

where d^f_i is the displacement vector of the discretisation points in the fluid domain, and v^g_i is the mesh motion velocity (i.e. $\hat{v}_i$). The mesh diffusion γ requires further parametrisation [22,23]. By a study of parameter dependency, one obtains an optimal approach concerning accuracy by choosing the diffusivity γ to be inversely proportional to the displacement.

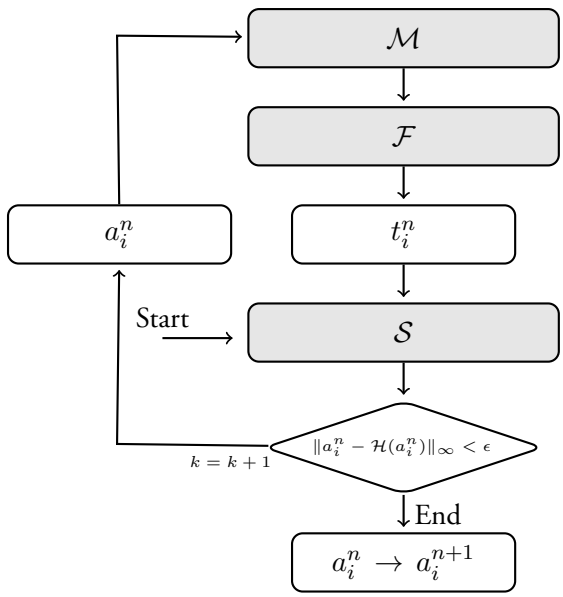

Figure 2.2: Partitioned FSI procedure: Coupling schema for a strongly coupled solution to Eqn (2.20). Grey boxes are stand-alone solvers, white boxes are the coupling interface.

2.5 The partitioned FSI methodology

With each stand-alone solver defined by Eqn (2.13) and the mesh motion solver $\mathcal{M}$, the FSI is formulated with a flow chart, as presented in Figure 2.2. The entry point to the iterative procedure is labelled as "Start" and the exit point as "End". The superscript n stands for the current time step.

The entry point ("Start") in the FSI is extrapolated, the predictor step, from the previous time step by applying velocity and acceleration for the displacement and/or regression extrapolation for the pressure ($k = 0$),

$$a_i^n \simeq a_i^{n-1} + v_i^{n-1}\Delta t + \frac{1}{2}v_{i,t}^{n-1}\Delta t^2 \ , \ p_i^n \simeq c_l p_i^{n-l}. \tag{2.19}$$

As explained in the forthcoming solution procedure, the predictor still applies for the displacement irrespective of the entry point. The shaded boxes represent the stand-alone solution step; the white boxes are the interfaces that integrate the separate solvers into an FSI solver. The Mesh motion solver for the fluid domain is normally integrated into the same software package as for the fluid. The Gauss-Seidel fixed-point problem can be expressed as,

$$a_i^n = \mathcal{H}(a_i^n) = (\mathcal{S} \circ \mathcal{F} \circ \mathcal{M})(a_i^n). \tag{2.20}$$

A solution is reached when with a sufficient number of subcycles k, the difference in norm between LHS and RHS is below a prescribed tolerance ($\|\mathcal{H}(a_i^n) - a_i^n\| < \epsilon$). The tolerance ϵ is set from a tolerance study as a function of the mesh size and time step.

2.6 Relaxation and acceleration techniques

Acceleration techniques originate from the study of accelerated convergence of sequences of scalars(a_1,...,a_n)[24] by applying a contracting map $\mathcal{T}$,

$$(a_1, ..., a_n) \rightarrow (\mathcal{T}(a_1), ..., \mathcal{T}(a_n)). \tag{2.21}$$

Of course, the mapping is such that the two sequences should have the same limit, $\lim a_i = \lim \mathcal{T}(a_i)$. One such of particular interest is Aitken δ^2 acceleration. By using the pseudo-inverse of a vector (i.e. $x_i^{-1} \approx \frac{x_i}{|x_i|}$), this method can also be generalised to handle a sequence of vectors. A more subtle approach to accelerate sequences is continuing the argument of Eqn (2.20) by reformulating the fixed-point problem in terms of a residual r_i^k at subiteration k,

$$r_i^k = \mathcal{H}(a_i^k) - x_i^k, \tag{2.22}$$

where x_i^k is the trial vector, then the Eqn (2.20) is said to converge as the infinity norm l^∞ of r_i^k reaches below a threshold, the FSI tolerance. The converged solution, i.e. the trial vector, is denoted as a_i^n, where n is the index for the current time step. It can be related to Newton's method, i.e. solving a root problem $\mathcal{R}(x) = 0$ with $\mathcal{J}dx \approx \delta r$, where $\mathcal{J}$ is the Jacobian matrix of $\mathcal{R}$. The Jacobian matrix can, however, be expensive to compute, but there exists an approach that allows combining the fixed-point iterative method with Newton's method in such a way that to a posteriori estimate the solution to Eqn (2.13) by using the sequence of the solution fields a_i^k for the current and the previous subiterations by the following assertion,

$$a_i^* = x_i^k + c_l r_i^l, \tag{2.23}$$

where $0 \leq l \leq k$. This procedure is also known in the literature as the Jacobian-Free Newton Krylov method (JFNK). A related technique is the successive under-relaxation method (SUR), originating from solving a linear system of equations[25]. This can further be related to a branch of different solution techniques.

These approximate solutions from Eqn (2.21) and Eqn (2.23) are applied to the mesh motion solver, except at the final subiteration to which a_i^n is used. For that reason, a_i^* is called the interface. From this, it is apparent that a kinematic synchronisation between the subdomains is attained only at convergence, and the choice of the epsilon is a critical step in determining the accuracy of the FSI solution. Since both the residual method above and the acceleration schemes act as a posterior correction, they are phrased as relaxation methods and form the relaxation step, which is denoted as $\mathcal{R}$.

The main drawback with partitioned techniques for FSI applications is the instability through the so-called added mass, which increases with increasing mass ratio. The relaxation resolves this. It should be noted at this point that acceleration techniques only partially resolve the instability.

2.7 Other FSI methodologies

In this section, other FSI methods that are of interest and mentioned in forthcoming chapters are defined here. Firstly, the focus is on techniques that differ by how the interface is moved and the choice of mesh representation. One notable of interest is the non-conforming mesh technique, immersed boundary technique (IB)[4], where one introduces a virtual force to redirect the flow around the structure, the mesh of respectively domain is overlayed, and the interface is often represented as Dirac kernels. The fluid is described by Eulerian description, and the solid is independent of the choice. A common issue with the IB method is the smearing, time stepping

restriction, and stability issues for larger bodies with more complex geometries[26]. This is the main reason that for a larger body, the conformed meshing is applied. Another category is the mesh-free technique, dealing with particle models in the Lagrange description. Two methods stand out in this category: Smooth Particle Hydrodynamic (SPH)[6] and Lattice Boltzmann (LB)[7]. The application area for SPH is wide, from large-scale astrophysical down to mesoscopic scales, such as free surface simulation with complex geometries with the effect of gravity and multiphase flows, allowing a cost-efficient procedure per cell count. The limitation is, however, the difficulty in boundary conditions or when the metric in space is not density. The space is discretised into a set of material elements with position and velocity, then the positions of the particles are computed by solving the motion equations, and the interaction between particles is achieved through kernel functions. Another limitation is that the amount of particles required increases with increasing resolution. Since the cost per particle becomes larger compared to mesh-based techniques, it comes to a tipping point where mesh-based methods are more accurate and faster. In the LB method, the space is discretised into a lattice of points (material element position), each connected to the surroundings by a predefined direction. Quantities along these directions define the fluid density. This method is well suited for parallelisation. Limitation of the model appears for high Mach applications, and there is also no current modelling for thermal-hydrodynamic problems. The Knudsen number governs the application range for the LB.

For cases of steady-state character, the dynamics of FSI can be attributed to the acceleration effect by the added mass. It can be possible, especially for structural-driven applications, to merely modify the mass matrix and solving the structural equation stand-alone, to estimate the frequency shift, induced stresses and amplitude of fatigue cycles, and this is then called the added mass technique[9].

Chapter 3

Literature study

The development of methodology in computational engineering is synchronised with hardware development. To illustrate this, a timeline of HPC development is presented with statistics to be used while discussing an overview of the FSI methodology. The statistics presented in this chapter, showing the techniques/years of publication and complexity/application presented, are taken from the FSI methodology related articles used as references in papers I-IV. Semi-empirical methods are excluded. The chapter ends with a summary of the articles used for the case setup, postprocessing, and validation as well as solution procedures. In addition, other methodologies not applied in this thesis are shortly discussed in order to create a context for the procedure chosen in this thesis.

3.1 Timeline of HPC and software package

Figure 3.1 shows the development of HPC hardware with the open-source software (relevant for the thesis) available for scientific computing and the performance peak in terms of the number of floating point operations per second [FLOPS] indicated in brackets. Around 1970 most of the keywords listed and discussed in the previous chap-

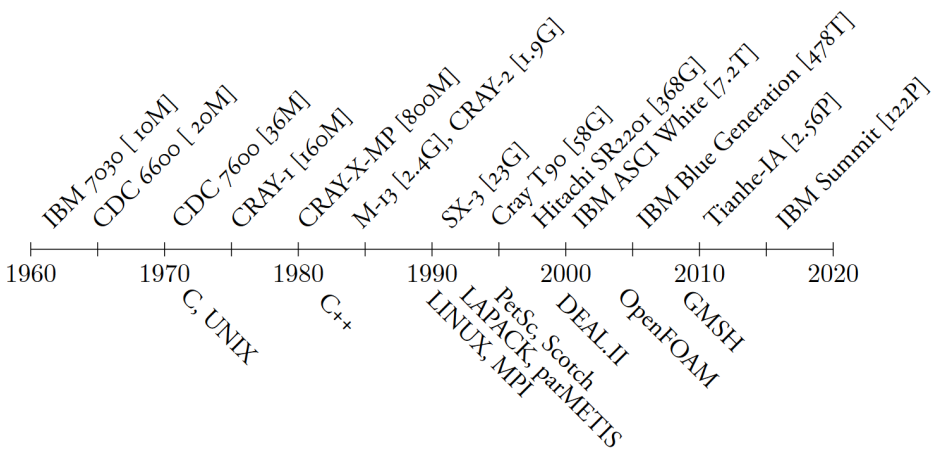

Figure 3.1: Timeline of HPC computer history of peak performance (above) [FLOPS], relevant package/language release date used in the research of this thesis (below)

ter were first developed: IB, ALE, SPH, FEM and INS solver. The terminology and even FSI methodology were presented, but the limited computational resources made the breakthrough only feasible once MPI protocol[27] and the software packages were in place around 1990 and 2000. Large software packages for FEA emerged in early 1960, for example, NASTRAN[28]. CFD software followed shortly with the development of techniques for solving the INS, more precisely, the projection methods for resolving pressure/velocity coupling by Chorin 1968[29]. An interesting aspect is that turbulence models were founded at an earlier point. Mostly since the need for development was based on the measurement techniques and using a statistical approach sidestepped the need for simulation of complex fields using computer resources. Take, for example, the Smagorinsky model from 1962[30]. Further, most FEA/CFD analyses at that time were done using pen and paper using scaling elements and framework/truss approaches. Using a computer at that time merely "copied" the handwritten procedures, developing algorithms purely in a black-box manner for computers was a concept not coined until 1970 and later. Take, for example, numerical recipes (1988)[31]. From 1980 to the beginning of 1990, they marked an era for black-box libraries such as BLAS[32] and LAPACK[33]. With the integration of HPC into the university, several pioneers laid the foundation for several large program packages around 1970 and 1980, such as ADINA[34] by Bathe, ANSYS by Swanson[35], Mathworks by Jack Little, Cleve Moler and Steve Bangert.

3.2 FSI complexity

The complexity of an FSI problem is defined by the mesh resolution required to solve a given problem and the order of accuracy, to which one coins the expression of the quantity and the quality of a method. However, complexity does not directly reflect efficiency and stability. Generally, the quality is the same between studies; hence, the complexity differs by quantity. The reported mesh size (i.e. the number of cells) on the year of published work is displayed in Figure 3.2. The data is further differentiated between two/three dimensional (2D/3D) mesh, and aeroelasticity (AE)/hydroelasticity (HE) is applied. The complexity increases with time, as to be expected as HPC computational power (FLOPS) increases. However, most articles report the usage of personal computers, and it is impossible to run the larger mesh without HPC ($\sim 10^6$). Despite the recent decades of improvements, there are sparse findings of studies of strongly coupled FSI using larger mesh; despite CFD breaking the billion limits, FSI reports up to 2020 presents seldom studies with a mesh of more than 10 M Cells.

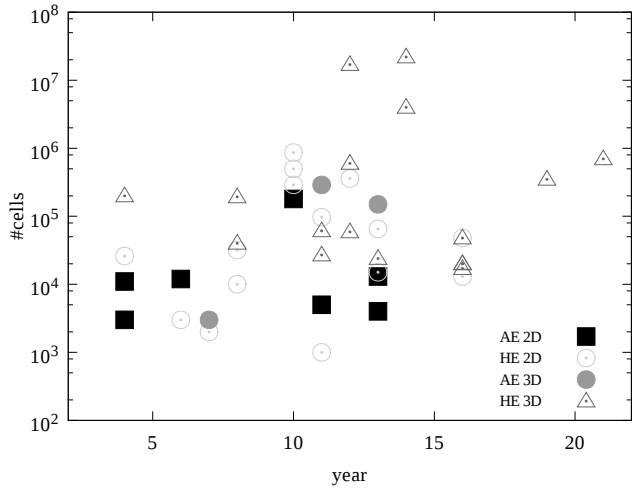

Figure 3.2: The complexity of FSI simulations over the past two decades. 2D/3D applications to hydroelastic (HE) and aeroelastic (AE). FSI solver applied in this thesis: Solving a 1 M cells fluid mesh, with 1000 macro steps with on average 3 FSI coupling iterations cost for an HPC that delivers 700 GFLOPS/node, about 20 hrs simulation time with two nodes.

3.3 Keyword distribution

For each FSI article used in the literature study, their solution techniques and applications were compiled and categorised, and their fraction was presented as pie charts, see Figure 3.3. The sections of the charts with a dark grey area show in which category the work of this thesis can be placed. The "Other" category represents a collective of keywords with a lesser amount than listed for the next smallest section.

The FSI problem can either be solved monolithically or partitioned. Semi-implicit is categorised as partitioned. The procedures for solving these are widely different; the most simplistic is the black-box principle, where only the stand-alone input/output are the transferred variables, whereas an interface program further stores the variables that can be processed. Another approach is to modify the stand-alone solvers to adapt to the FSI by computing the coupling elements in the FSI problem, a semi-implicit approach. For a monolithic procedure, the whole mathematical framework is unified and solved in one solver step. However, this is often not feasible, partly due to the different material settings and/or computational resource limitations. Independently, one can choose between different meshing, conforming or non-conforming mesh, leading to the differentiation between two different types of mesh handling: Arbitrary Lagrange (ALE) and immersed boundary (IB), the "Other" category contains one-dimensional/spectral approximations, often for benchmark and stability analysis, involving simplified governing equations, leading to either fluid or solid response to

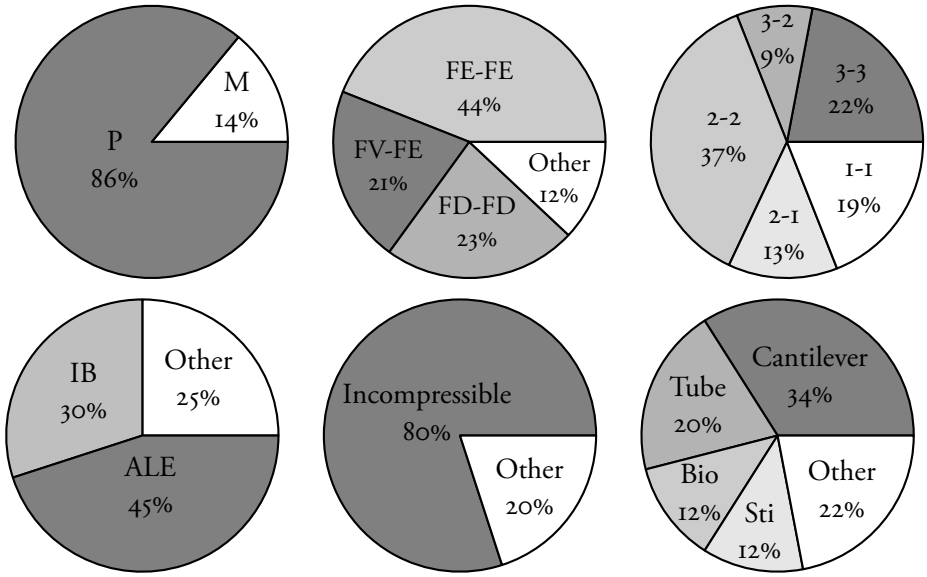

Figure 3.3: Upper - Left: Solution strategy. Middle: Discretisation procedure. Right: Dimension. Lower - Left: FSI description. Middle: Incompressible flow condition. Right: Application. Note: Bio=Biological, Sti=Still tank

be one dimensional, so then ALE/IB loses its purpose. The governing equation can then, on each mesh, be solved by Finite Element/Finite Volume and Finite Difference (FD); the "Other" category contains methodologies such as SPH/LB and other combinations of discretisation; one such particular of interest is FV-FV, despite its popularity in some applications, the actual influence is much lesser on research forums, mainly due to the difficulty in representing the material description for a solid in FV framework. The spatial dimension of the mesh is either one/two/three dimensional, but a surface in three dimensions is represented as two-dimensional and so forth. Therefore, the combined FSI problem may differ in the dimensionality of the mesh. The spectral method is also common but mostly as a procedure to estimate stability/efficiency, applied to one-dimensional problems (1D) to provide analytical expressions, but it is placed under "Others" among different combination techniques. What dimension/discretisation on each mesh is specified with a hyphenation starting with the fluid and then the solid.

There is a missing dimension: 3-1, which commonly appears in Vortex-induced vibration (VIV) applications. We are also excluding all references where the dimension of the solid is higher than that of the fluid since that often deals with simplified FSI, such as added mass methodology.

The two final keywords of interest are the incompressible condition on fluid, whose

"Other" represents inviscid and compressible. The exclusion of solids is due to the difficulty in modelling other than compressible. Secondly, for applications, the cantilever dominates, followed by tube (pipe, shock tube, thin membrane pipe, channel) where either the whole/part of the wall is flexible; then comes biological applications (blood vessel, heart, wings), followed up by still tank applications (moving lid on the container, tilting liquid tanks partitioned by a membrane/plate). The "Other" category contains mostly one-dimensional applications such as piston, spring-dash pod, thin-membrane, and immersed cantilever in inviscid, primarily to estimate the stability and determine the added mass effect and efficiency analysis. There are cross-overs with the still tank, but the other keywords take precedence.

3.4 A review of previous work on the subject and application

Firstly, the background and the application will be reviewed, and some terminology will be defined to be used in chapter Results. Secondly, the context in which the FSI methodology developed in this thesis will be reviewed. It should be noted that the Finite strain theory was well-defined even before the 1950s, somewhere between the birth of continuum theory in 1878 when George Cantor began to work through Hilbert's question to the book of Truesdell[36] or for a more modern framework[37,38]. A similar argument goes with incompressible Navier Stokes from around 1850 to the current date[39,40]. With the theory for the solid and fluid set to stone, the research is focused on the challenge and the difference in implementing the mesh motion strategy and relaxation together with action steps. To avoid bloating the remaining sections, a chosen few are presented, ensuring references therein cover all relevant articles used in this thesis.

Application: immersed beams in quiescence and channel flow

Sarpakaya[41], provides an excellent review on the subject of Vortex-Induced Vibrations (VIV). The flow characteristic generated by VIV in the application of bluff bodies has been in focus for several decades for various reasons: from basic research of physical phenomena, e.g. wake structures or parameter dependencies, towards engineering applications with specific goals, such as reduction/maximisation of lift and drag effects. Other areas of interest are graphics and visualisation for the purpose of providing realistic flow animation. A bluff-body and streamlined body are often characterised by the drag and the lift coefficients, which are key design factors for the engineer. See, for example, the work of Sumner[42] on the clamped-free short AR cantilever, or drag/lift coefficient dependency on various flow characteristics by C.Norberg[43]. Most applica-

tions for VIV are offshore type; the focus is therefore much centred around long and slender cylinders, and the shedding modes downstream are often characterised as two-dimensional in cross-flow/streamwise with oscillation force cross-flow, see the works of Bearman[44] and Williamson[45]. These modes can be parameterised as a function of the Reynolds number and velocity of the free stream flow. Recording the pattern/-mode as a function of forced amplitude/frequency for a given flow speed presented a map with distinct regions for these patterns to emerge. Two-dimensional shedding modes for non-deforming cylinders, forced into oscillation cross-flow, are often distinguished between 2S, 2P, 2C, 2T, S+P, 2P+2S, P modes[46] and non-synchronised. Similar to flow over plates, hairpin vortices appear aligned streamwise, appearing with a regular shape and equidistance between them as a function of the diameter of the cylinder, one of the criteria for classification of these modes, notable two modes, firstly appearing around 200 Reynolds number, then transition into B appears around Re 400, other modes exist, but mode B exist in transition regime for various applications, such as clamped-clamped cantilever, flow over splitter blade and so forth. In addition, near corners and free-end, so-called trailing vortices appear, which shape the wake behind the bluffbody, notable for sharp corners[47]. For free-end regular-shaped bluff bodies, two additional distinct flow phenomena shape the wake, the downwash and the upwash. These are best characterised as redirected flow from below around the bluff body upwards behind the body, leewards, and, across the free-end downwards. respectively.

The measuring techniques were often limited to hotwires/instant snapshots with smoke as visualisation, whereas the cylinders often were thin wires/slender tubes. Three-dimensional flow characteristics/modes[47], to describe these, one often depicts mean field flow, and from there, recirculation zones are identified. These are observed as strongly dependent on the aspect ratio (height to frontal width)[48], free-end[49] or clamped-clamped and the flow speed. Therefore, the position of these centres, radius, and topological flow pattern are often used to characterise the settings of an experiment. Dimensional analysis shows that reduced flow speed $U_R = \frac{U}{fD}$ with the frequency of the forced oscillation (f) and cylinder diameter (D) scaling with the flow speed (U) is a governing variable for VIV, and research outcomes often characterise the results in terms of these by presenting reduced amplitude (forced) versus reduced U_R or Strouhals number. By having a cylinder hinged with springs and restricting the motion to the cross-flow direction, a non-forced motion can be achieved. Note that the the reduced velocity in this case includes the eigenfrequency of the system. For this kind of motion, the displacement as a function of reduced velocity exhibits a pattern that can be divided into three regions: pre-synchronized, synchronized and desynchronized. Comparing this to the forced non-deforming, the pre-synchronised is mostly located in the 2P/C(2S) and 2S region, the synchronised mostly in 2P and desynchronisation in the same region. However, as shown by Vikestad, there can be

significant differences between free/forced motion[50]. The difference between elastic-mounted and flexible cantilevers in three-dimensional is an extra mile that is easily overseen. Despite that, there are strong similarities in the output, mostly because the reduced response graph is one of the strong features within the VIV community. Moreover, a distinct phenomenon known as the upper/lower branch, a hysteresis effect, is also observed while decreasing and increasing flow speed[51]. Elastic-mounted non-deforming cylinders, allowing streamwise flexibility, have been in focus, giving similar response patterns, different in quantity, but still the same pattern. Fully flexible and large deformation is mainly restricted to laboratory-scaled studies related to seaweed response patterns[1] and fin/foil motion[52] to understand the mechanism of surrounding flow. Instantaneous flow pictures are mostly depicted in articles to uncover flow topology and their origin, but in practice, they hold a limited value from a scientific point of view, reproducibility. Different shedding mechanisms have been proposed for the above scenarios; despite that, the wake ought to be similar, and the following inherent Hoph-instability and non-linearity make the classification and modelling exceptionally difficult[53].

There are numerous more articles on this subject, but omitted here. The main reason is that the goal is on the methodology, not the application. As the reader may already observe, the keyword turbulence is omitted, but that is covered within the reference above.

FSI methodology

For a more rigorous approach, the interested is referred to some of the reviews on this subject, for the coupling scheme procedures[54], IB procedure[26], strangely enough, no review on ALE could be found, but numerous rigorous works exist on the subject such as the work by Hirt[55] or Donea et al.[2]. For relaxation techniques, there are several different, but within the FSI community, notable IQN-ILS by Degroote[56] or Aitken relaxation δ^2[57] by Küttler. The JFNK methodology by Knoll[58], which relates to Anderson mixing[59], and in turn is closely related to IQN-ILS, provides a whole branch of different approaches. There is an interesting investigation, saying that Anderson mixing is best suited for first order convergence[60]. An outlook on partition methodology and different coupling strategies was presented by KC Park[61], and from one of its references within, there is a notion about SSI that subsequently led to the validation case in paper I. The earliest FSI methodology closest to the presented in this thesis is the work by Wood[18], essentially the same main steps with the difference of usage of the Finite Strain methodology present in the work by Bathe[62]. One stabilisation procedure that made the methodology feasible before JFNK relaxation methods were implemented was the continuation technique; there is a corresponding work

about fictitious mass/damping procedure that applies a similar approach[63], which is applied implicitly to other techniques, such as space-time discretisation[5]. There are several similar methodologies that can be traced back to the early 60s, such as the method of lines[64] or false transient analysis. Monolithic solvers are quite commonly seen in the literature, for example, the original work by Peskin 1972[4] using IB procedure. Of course, there are monolithic solvers based on ALE[65]. However, monolithic solvers are primarily used in comparison to partitioned[66], and there are sparse reports with application to larger FSI problems. In the study of a GMRES-based monolithic FEM solver by Heil[67], it was concluded that the dimensionality issue made monolithic challenging to handle since it required sufficient high resolution and stabilisation to provide accurate results, also providing insight into the importance of ensuring the accuracy by studying the pressure response. Otherwise, it may produce a substantial pressure difference, leading to inconsistent solutions. Although this observation applies to the Galerkin procedure with fluid solver using SUPG stabilisation and multigrid, it is sensible to assume that any iterative procedure would show similar sensitivity. Further, DEAL.II supports dual-weighted residual functionality with error indicators and orto mesh refinement, but it is not used in this thesis. The FEM-FV partitioned solver is used because most solvers often apply pure FEM-FEM or FV-FV (60%).

Using a structured grid in the solid and an unstructured grid in the fluid raises concerns regarding the impact of interpolation error at the surface. A study by Jaiman et al.[68] shows that point-to-point interpolation introduces spurious oscillation that leads to inconsistent solutions, notable for shock expansion. When measuring the quality of a methodology, one often measures it by the quantity of cells in a mesh required to reach a certain accuracy. The TVD scheme in OpenFOAM[69] is described as reproducing "quite accurately the three different exact solutions" but observed somewhat larger dissipation and thus requiring a larger mesh, especially while using unstructured mesh. Still, quantity trumps quality; it is sometimes more effective to have a higher resolution of the mesh to ensure the error is sufficiently low than to improve the quality of methodology. In addition, using unstructured mesh on the fluid and structured mesh in the solid generally requires interpolation, and one of the most commonly used is the inverse interpolation distance weight methodology (IDW)[70], which usually lowers the accuracy. Therefore, a mesh dependency study is necessary for every simulation to assess the introduced errors.

In the choice between ALE and IB methods, whenever both are eligible for the same application, the IB approach is often first reported, but turbulent flow limits the usage of IB, leaving room for the more complex approach with moving mesh, the ALE approach. The debate over which mesh motion approach is the most suitable in the literature is as infectious as which discretisation is most suitable on FSI. For example,

a related technique to IB is the fictitious domain method[71] but adapted to FD, or an IB versus ALE with application to heart valve[72], with opposite results, and that is due to a common line, the application, and re-meshing. The ALE approach is more accurate or efficient as long as re-meshing is unnecessary. Nevertheless, it should be remembered that the ALE procedure has issues with the LES approach[16,73]. Another issue often recurring for ALE is the GLC error; however, for FV methods, it has less impact[19]. An interesting but different procedure is the SPH solver[6,74], a mesh-free Lagrangian based methodology, which has difficulty in handling turbulence and boundary conditions as well as lower efficiency for higher viscous fluid problems.

Numerous two-dimensional in-house FSI procedures have been reported; one of particular interest is the work of Turek[75], using a monolithic IB solver, providing a benchmark test that most of the ALE mesh motion solvers cannot reproduce without re-meshing. Then there is a 3D elastic tube shock tube[76], that is gently put aside; it is an interesting benchmark but not applicable with an incompressible solver without correction terms in discretisation to the openings and mysterious justification. For turbulent flows, using LES, one cannot apply two-dimensional validation cases and expect reproducing expected behaviour such as scaling in kinetic energy. For that reason, two-dimensional studies have been removed from this study and, therefore, not further discussed. However, with immersed flexible cantilever with large deformation as an application, there are almost no three-dimensional applications to cantilever, with few exceptions, such as the work by Tian et al.[77] and Wang[78], and Nayer et al.[79]. The application of interest in the work of Tian et al. is with an application to submerged vegetations by Luhar and Nepf[1] using the IB approach and under-relaxation to stabilise the strongly coupled. Nayer et al. use ALE methodology to a partitioned technique stabilised by under-relaxation applied to a splitter blade, a three-dimensional form of the Turek benchmark test. Also, Nayer et al provide detailed experimental data for the same application, showing a good match toward simulation. Unfortunately, their case setting was not feasible for the settings of the mesh motion for OpenFOAM at that time. Not many have applied this splitter blade case compared to the work by Luhar[1]. Commercial programs that handle FSI exist, such as ANSYS-FLUENT, which have been used in numerous different reports[80,81], but one of interest is ANSYS-CFX[82]. Also, ADINA is commonly applied[83,84]. However, the advantage of partition techniques arises from the fact that with an interface program package, one can use different programs to handle the coupling procedure. Several interfaced open-source codes existed, for example, MPCCI[85], PreCICE[86], Elmer[87].

A method's efficiency is often measured by the wall time required to accomplish a simulation and its stability by the convergence radius of the method's governing parameters, namely, the mass ratio and form factor[88], which also can be related to the

added mass effect and frequency shift. There are notable differences between the conclusion of stability and efficiency and the choice of relaxation, but some aspects are common between studies. For example, in Quasi-Newton residual techniques, 50% gain in efficiency is to be expected[89], some notable parameter dependency study has been reported[9,90,91], showing that with increasing temporal order or increasing mass ratio, the stability decreases.

Chapter 4

Solution procedure

This chapter presents the solution procedure for each solver, the FSI coupling strategy, and parallelisation. It begins with presenting the specific procedure used as a fluid step solver, which is used as a black-box obtained from the OpenFOAM package[92]. Similarly, but somewhat in more detail, the solid step solver is outlined, mostly since there is no black-box solver available within the deal.II package[62]. Then the chapter ends with a thorough algorithm presentation of all relaxation methods applied. Appendix A and B outlines terminology for different discretisation approaches, FVM and FEM.

4.1 Fluid solver step ($\mathcal{F} \circ \mathcal{M}$)

The $\mathcal{F}$ and $\mathcal{M}$ are described within the same formalism, the Finite Volume Method (FVM). The package used is OpenFOAM[93] and this section outlines the solution procedure using FVM. The entry point is the balance equation for a control volume, a volume integral representation. The FVM essentially transform the volume-based integrals of balance equations into face integrated ones with a summation of fluxes that is conservative. See further in any course book on the subject[40,94] for more details of terminology applied and formulae applied.

4.1.1 Solving INS step ($\mathcal{F}$)

The INS equation, namely Eqn (2.5), consists of a temporal term, a Laplacian term (diffusion), a SGS term (turbulence modelling), a gradient and a divergence term (convection). Now, set $\phi = U_i$ and each term is evaluated separately, and added together to give the local balance equation.

The divergence term in the INS can be identified as a contraction of the mass flux term over facets ($F = S_j(\rho U_j)$) with face values ϕ_f of ϕ,

$$\int_{CV} (\phi \rho U_j)_{,j} dV = \int_{\partial CV} n_j (\rho U_j \phi) dA = F_k \phi_f^k, \tag{4.1}$$

where k is the face index, and S_j is surface area normal.

Likewise, the gradient of the pressure,

$$\int_{CV} p_{,i} dV = \int_{\partial CV} p n_i dA = (S_i)_k p_f^k, \tag{4.2}$$

where k is the face index. The Laplacian term, expressed as the divergence of the gradient of U_i, then becomes

$$\int_{CV} \nu \phi_{,jj} dV = \nu \int_{\partial CV} n_j \phi_{,j} dA = \nu (S_j)_k (\phi_{,j})_f^k, \tag{4.3}$$

where k is the face index. In the closure of the SGS residual tensor (B_{ij}), the Smagorinsky-Lilly approximation is applied, using the rate strain (D_{ij}),

$$B_{ij} = -2\nu_{\mathrm{SGS}} D_{ij}. \tag{4.4}$$

where the SGS viscosity is calculated from

$$\nu_{\mathrm{SGS}} = C_s \Delta |D_{ij}|. \tag{4.5}$$

Hence, an additional Laplacian term, leading to modification of the viscosity ν into an effective viscosity $\nu_{eff} = \nu + \nu_{SGS}$.

It should be noted that the value of the Smagorinsky coefficient[95] (C_s) is not universal and that this model often overestimates the dissipation. An alternative approach is to apply a least square fit in which one obtains an explicit expression for C_s in terms of the velocity field and a test filter, the Lilly Model[96]. This approach is with limited application because it uses a homogeneous approximation in the average procedure, i.e. the averaging along each principal direction. Such approximation is suitable for flow with little obstruction and flat surfaces. For all simulations, the tophat filter is applied as a test filter. This approach also includes the transfer of energy from the subgrid to the resolved scale, the so-called backscatter. However, in the current implementation of OpenFOAM, the subgrid viscosity is limited to positive values for numerical stability.

By summation of each term, assuming the cell size to be fixed, a semi-discretised form of INS is obtained for each CV, i.e. keeping the derivative over time non-discretised, with an RHS equal to zero (no source),

$$\begin{aligned} \int_{CV} (\phi_{,t} + (\phi U_j)_{,j} + \nu \phi_{,jj} + p_{,i}) dV = \\ \phi_{,t} + F_k(\phi)_f^k + (S_f)_k p_f^k + \nu (S_j)_k (\phi_{,j})_f^k = \\ \phi_{,t} + a_P \phi_P + \sum_N a_N \phi_N, \end{aligned} \tag{4.6}$$

where the summation is over N, a_P, and a_N are the coefficients gathered for respective source nodes, owner CV and neighbour cells. The final term is the time evolution, to which the backward interpolation scheme is applied,

$$\phi_{,t} = \frac{1}{\Delta t}(\frac{2}{3}\phi^n - 2\phi^{n-1} + \frac{1}{2}\phi^{n-2}), \tag{4.7}$$

which is second-order accurate over time. Adding this to Eqn (4.6) and moving around the terms, gives a non-zero RHS phrased as R_j and the local discretisation is finished, providing the local matrix ($A^e_{ij}\phi^e_j = R^e_j$)[97]. Using the localisation relation $(Z_e)_{ij}$, i.e. Eqn (2.11), the global degrees of freedom ϕ_i is mapped to the local ϕ^e_i, i.e. the global discretisation ϕ_j by incrementally summation,

$$\sum_{CV} A^e((Z_e)_{ij}\phi_j)_l = \left[\sum_{CV} A^e((Z_e)_{ij})\right]\phi_j \equiv A_{ij}\phi_j, \tag{4.8}$$

which forms the desired solver as a matrix A_{ij},

$$A_{ij}\phi_j = R_i. \tag{4.9}$$

This is the assemblage stage for the global representation of the momentum equation.

However, the pressure field is unknown. To resolve this, a projection method is applied called the PISO algorithm[98], which formulates a pressure equation to enforce mass continuity, then applying a predictor-corrector procedure. In PISO one begins with a guess of pressures and fluxes, then solves the momentum equation, and then by applying divergence of momentum, reformulates the equation into a pressure equation using intermediate fields. To achieve continuity to be satisfied the procedure reiterates until convergence of the involved fields. To minimise the influence of checkerboard in pressure by using a collocated grid, Rhie-Chow interpolation is applied, where the gradient of pressure/velocity is computed from face values.

Together, the procedure to determine the pressure is phrased as the Rhie-Chow PISO algorithm. To outline this procedure one first separates the contribution from the owner CV indexed P from the neighbour N field,

$$a_P(U_i)_P = -\sum_N a_N(U_i)_N - \frac{1}{V_P}p_{,i} \tag{4.10}$$

Then prior definition of intermediate velocity, the H_i field is defined,

$$H_i = -\sum_N a_N(U_i)_N, \tag{4.11}$$

the Rhie-Chow interpolation step begins with,

$$(U_i)_P = \frac{H_i}{a_P} - \frac{1}{a_P V_P} p_{,i} \tag{4.12}$$

by substituting Eqn (4.11) into Eqn (4.10). Now the continuity condition for incompressible flow, followed by Eqn (2.4), leaves the LHS zero for INS, defining the intermediate velocity field as $U_i^* = \frac{H_i}{a_P}$ gives the following local discretisation of pressure,

$$U^*_{i,i} = \frac{p_{,ii}}{a_P V_P}. \tag{4.13}$$

Then one forms the global equation by assemblage. Note that multiplying by V_P on both sides $U^*_{i,i}$ is changed to a volumetric flux ϕ^*. Further, the mass flux is computed differently for INS using this procedure, see section 3.8 in Jasaks thesis[97]. This intermediate velocity term makes it necessary to iterate. To summarise the PISO step, with the velocity field, the momentum equation, and guessed pressure (previous time step),

- Compute H_i : Eqn (4.11).
- Compute the mass flux on the cells faces
- Solve the pressure equation : Eqn (4.13).
- Compute the $(U_i)_P$ and mass fluxes : Eqn (4.12).
- Update boundary conditions.
- Repeat from the start for a prescribed amount of iterations.

There is an additional extension of the PISO at this stage, the PIMPLE algorithm, which merely recomputes the momentum equation (outerloop) and re-enters the PISO with under-relaxation (inner loop). This forms the fluid solver $\mathcal{F}$ that is being applied. It should be noted here, that the actual necessity for the outer loop is merely to allow much larger time steps, suitable in FSI, the convergence of the velocity field is ensured by convergence in the pressure field from the inner loop.

The OpenFOAM package provides this solver and is used as a black-box solver expressed in non-dimensional units similar to Eqn (2.6). The choice of discretisation, interpolations, correction, and values of correction/blending are specified by keywords of format "<discretisation> <interpolation> <correction> <values>". The non-temporal terms that need discretisation use Gauss scheme, and the temporal uses "backward". The interpolation for non-divergence terms, uses "linear", while the divergence term "limitedLinear" (TVD scheme). What is not discussed here but can

be found elsewhere, is the correction of the non-orthogonality in the mesh, see for example Jasaks thesis[97]. The motion is integrated into the momentum equation by replacing the convection U_i field with the relative velocity field with respect to the mesh motion field obtained from solving the mesh motion equation Eqn (2.18). This introduces an additional field to be computed, but it is not part of the FSI boundary condition, the mesh velocity. However, the motion solver has been separated code-wise and treated as a stand-alone solver and thus is described separately next section.

The turbulence model described above does not account for the transported sub-grid turbulence, which is then added as a transport equation of SGS kinetic energy $k = \frac{1}{2}\overline{u_i u_i}$. The transport of sub-grid kinetic energy can be obtained from the filtered INS Eqn (2.10), and by introducing the modelling described above, also by neglecting the viscous large-scale dissipation, the transport equation for the variable k is defined by Yoshizawa and Horiuti in[99] as,

$$k_t + (k\bar{U}_i)_i = \nu_{\text{eff}} k_{,ii} - B_{ij} D_{ij} - C_s \frac{k^{3/2}}{\Delta}, \tag{4.14}$$

where $\nu_{\text{eff}} = \nu + \nu_{\text{SGS}}$ and ν is the kinematic viscosity. Eqn (4.14) is solved as a post-correction, after the resolved variables for a given time step, are computed. This approach is superimposed with Lilly's model[99].

4.1.2 The Mesh Motion solver ($\mathcal{M}$)

The diffusion equation Eqn (2.18) requires a definition of the diffusivity. By setting

$$\gamma = \frac{C}{r_c^n} \tag{4.15}$$

where r_c is the distance from the centre to the nearest moving boundary cell centre. The coefficient C is a scaling factor concerning the initial values that can be further parametrised. This leads to the following discretised form of the motion equation,

$$S_j^k \frac{C}{r_c^n} (d_{i,j})_f = 0. \tag{4.16}$$

Moreover, the prescribed boundary moved to the RHS sets the equation to be solved. However, while deforming the cells, the continuity is violated. The Geometric Law of Constraints (GLC) must be satisfied. This is obtained by rewriting Eqn (2.4) and then apply the incompressibility condition ($U_{i,i} = 0$),

$$\frac{\partial}{\partial t} \int_{CV} dV_f = \int_{S_f} v_i^g n_i dA = S_i^k (v_i^g)_f^k. \tag{4.17}$$

This equation is be solved in two steps. Firstly by solving Eqn (4.16), then from the corresponding mesh displacement from the previous time step, compute the change in volume by Eqn (4.17) by the RHS computing face velocity for each face, averaged face area. One then applies Euler forward stepping on the temporal term on LHS, multiplied together with time step gives the swept volume, by division of swept area, distance swept along face normal is computed (δX). This gives a provisional expression for the mesh velocity v^*,

$$v^* = \frac{1}{\Delta t} \sum_f \delta X_f. \tag{4.18}$$

Then, with these stored provisional velocities, one adopts the same discretisation scheme for the momentum equation to ensure proper time marching at the coupled boundary. The time marching scheme on LHS is the same temporal discretisation used for the INS. The GLC is satisfied as one introduces the aforementioned relative velocity U_r into the INS. Similarly as with INS, using OpenFOAM, the above solution procedure is specified by setting the keyword for the solver as "displacementLaplacian" and the metric of diffusivity "inverseDistance".

4.2 The Solid step ($\mathcal{S}$)

First the equation of motion Eqn (2.7) is transformed into a weak form, expressing the virtual work in the current configuration in terms of the virtual displacement (δa_i) and Voigt notation with operator matrix L_{ij},

$$\int \delta a_i (L_{ij}\sigma_j + f_i^e - \rho \ddot{a}_i) dV = 0, \tag{4.19}$$

which after integration and introduction of shape functions (H_{ij}) gives the discretised form of the equation of motion,$a(\xi) \rightarrow H_{ik}(\xi)a_k$, where a_k is a discrete vector over local space). After applying the divergence theorem, moving partial derivatives of the stress to the displacement, strain (ϵ_i) is introduced. Moving force contribution to the right and keeping the acceleration on the left the equation of motion then becomes

$$M_{ij}\ddot{a}_j = f_i^e - f_i, \tag{4.20}$$

where M_{ij} is the mass matrix and f_i the internal force ($f_i = \int B_{ij}\sigma_i dV$). For the details and notations, see further in any course book on the subject[37,38]. The strain displacement matrix is $B_{ij} = L_{ik}H_{kj}$. Also, g_i is the acceleration of gravity.

4.2.1 Constitutive relation

To relate the strain with stresses, constitutive relations must be defined, here focusing on reference configuration (X). As such, the hyperelastic material description is applied, where one assumes a potential function ($\mathcal{W}$) from which the nominal stress can be evaluated, the Second Piola-Kirchhoff tensor S_{ij},

$$S_{ij} = \frac{\partial \mathcal{W}}{\partial E_{ij}}, \tag{4.21}$$

where Green strain tensor E_{ij} is defined by,

$$E_{ij} = \frac{1}{2}(\frac{\partial a_i}{\partial X_j} + \frac{\partial a_j}{\partial X_i}) + \frac{1}{2}\frac{\partial a_k}{\partial X_j}\frac{\partial a_k}{\partial X_i}. \tag{4.22}$$

The Saint-Venant-Kirchhoff model is applied,

$$\mathcal{W}(E_{ij}) = \frac{1}{2}D_{ijml}E_{ij}E_{ml}. \tag{4.23}$$

Using the isotropic material assumption, D_{ijml} can be defined as

$$D_{ijml} = \lambda\delta_{ij}\delta_{ml} + \mu(\delta_{im}\delta_{jl} + \delta_{jm}\delta_{il}) \tag{4.24}$$

where the Lamé parameters λ and μ are related to Young modulus (E) and Poisson ratio (ν_s) as

$$\lambda = \frac{\nu_s E}{(1+\nu_s)(2\nu_s - 1)} \tag{4.25}$$

$$\mu = \frac{E}{2(1+\nu_s)}. \tag{4.26}$$

The internal force is, however, a function of velocity and displacement, see Eqn (2.7), which in discretised form phrased as $\mathcal{F}$, is a reminder of this, but a linear split between velocity and displacement is assumed,

$$\mathcal{F}(a_{i,t}, a_i) = \mathcal{D}(a_{i,t}) + \mathcal{G}(a_i), \tag{4.27}$$

where $\mathcal{G}(a_i)$ is the internal force for a static case and $\mathcal{D}(a_{i,t})$ is the damping term, which need further modelling.

4.2.2 Finite strain and total Lagrangian formulation.

Assuming static case, dropping the temporal terms, set the time evolution as $S_{ij}^{t+\Delta t} = \Delta S_{ij} + S_{ij}^t$ the virtual work, Eqn (4.19) can be expressed as,

$$\int_{CV} \delta E_{ij}\Delta S_{ij} dV + \int_{CV} \delta E_{ij} S_{ij}^t dV = \int_{CV} \rho\delta a_i g_i dV + \int_{\partial CV} \delta a_i (t_0)_i dA, \tag{4.28}$$

where t_0 is the nominal traction, i.e. the traction acting in the reference configuration. Now, using the constitutive relation between strain and stress on the increment of $S_{ij}^{t+\Delta t}$, (i.e. $\Delta S_{ij} = D_{ijkl}\Delta E_{kl}$). As the strain increment due to Eqn (4.22) contains linear (ϵ_{ij}) and quadratic (η_{ij}) terms, i.e. $\Delta E_{ij} = \Delta e_{ij} + \Delta\eta_{ij}$, and its variation ($\delta E_{ij} = \delta\Delta E_{ij}$), applying this to Eqn (4.28) and and omitting second order and higher order terms with respect to increment gives,

$$\begin{aligned}\int_{CV} \delta\Delta e_{ij} D_{ijkl}\Delta e_{kl} dV + \int \delta\Delta\eta_{ij} S_{ij}^t dV = \\ \int \delta a_i(t_0)_i dA + \int \rho\delta a_i g_i dV - \int \delta\Delta e_{ij} S_{ij}^t dV.\end{aligned} \tag{4.29}$$

By rearranging terms on LHS, an operator matrix L_{ij} can be defined for the purpose to introducing Voigt notation (i.e $\Delta e_i = L_{ij}H_{ij}\Delta a_i$). However, for finite strain, the strain displacement matrix B is a sum of a linear part (B_L) and non-linear (B_{NL}). Introducing $B_L = LH$ then $\Delta e_i = (B_L)_{ij}\Delta a_j$ and by defining $K_L = \int B_L D B_L dV$ one reformulates first term on LHS as

$$\begin{aligned}\int_{CV} \delta\Delta e_{ij} D_{ijkl}\Delta e_{kl} dV = \delta\Delta a_i \left[\int (B_L)_{ik} D_{kl} (B_L)_{lj} dV\right] \Delta a_j = \\ \delta\Delta a_i (K_L)_{ij}\Delta a_j.\end{aligned} \tag{4.30}$$

The linear part of the discretisation is obtained. The second term in LHS, the non-linear part, can be similarly abbreviated as $\int \delta\eta_{ij} S_{ij}^t dV = \Delta a_i (K_{NL})_{ij}\Delta a_j$. The so-called geometric contribution is defined by $K_{NL} = \int B_{NL}^t \mathrm{T} B_{NL} dV$, where B_{NL} are the non-linear strain matrix respectively shear matrix T. Altogether, one obtains the following discretised version for the virtual work with finite strain for the static case,

$$\delta\Delta a_i (K_L + K_{NL})_{ij}\Delta a_j = \delta\Delta a_i (f_i^{e,t+\Delta t} - f_i^t). \tag{4.31}$$

with the external (f^e), sum of the acceleration gravity (g_i) and the traction (t_i), computed in discrete space by introducing H_{ij}, and the internal forces (f) on RHS, computed as $\int BS dV$. The linearised displacement increment then becomes

$$(K_L + K_{NL})_{ij}\Delta a_j = (K_t)_{ij}\Delta a_j = f_i^{e,t+\Delta t} - f_i^t \tag{4.32}$$

As stated above, the tangential stiffness matrix K_t is applied to solve the problem by an increment solution procedure. However, at this point, temporal discretisation must be taken into consideration.

4.2.3 Temporal discretisation

The semi-discretisation form of Eqn (2.7) takes the following form,

$$M_{ij}\ddot{a}_j^{t+\Delta t} + \mathcal{F}(a_i^{t+\Delta t}, \dot{a}_i^{t+\Delta t}) = f_i^{e,t+\Delta t}, \tag{4.33}$$

where $\mathcal{F}$ is the internal force. Taking the residual r, being the difference between the RHS and LHS, then using the chain rule the variation of the residual becomes,

$$\delta r_i = \frac{dr_i}{da_k} da_k = \frac{\partial r_i}{\partial a_k}\delta a_k + \frac{\partial r_i}{\partial \dot{a}_k}\delta \dot{a}_k + \frac{\partial r_i}{\partial \ddot{a}_k}\delta \ddot{a}_k. \tag{4.34}$$

The first and last term can be identified from above as $-(K_t)_{ij}$ and $-M_{ij}$ respecitvely, and define $C_{ij} = -\frac{\partial \mathcal{F}}{\partial \dot{a}}$. The Newmark algorithm computes increments from acceleration using the integration of parts which takes the following form

$$\delta \dot{a}_i = \gamma \Delta t \delta \ddot{a}_i \tag{4.35}$$

$$\delta a_i = \beta \Delta t^2 \delta \ddot{a}_i \tag{4.36}$$

giving the final expression

$$\delta r_i = -\left[K_t + \frac{\gamma}{\beta \Delta t} C + \frac{1}{\beta \Delta t^2} M\right]_{ij} \delta a_j = -(K_t^*)_{ij} \delta a_j, \tag{4.37}$$

leading to

$$(K_t^*)_{ij} \delta a_j = r_i, \tag{4.38}$$

since $r_i^{t+\Delta t} = r_i + \delta r_i \approx 0$. This allows for an incremental solution approach, a conventional Newton-Raphson method. The only unknown term is the C_{ij}, which represents the damping in the material and thus requires constitutive relation towards governing variables.

4.2.4 Rayleigh damping model

The C_{ij} term in Eqn (4.37) is computed from Rayleigh damping model, which is obtained by an ad-hoc assumption of prescribing a linear dependence in the velocity $\mathcal{D}(a_{i,t})$ giving the following expression,

$$\mathcal{D}(a_{i,t}) = C_{ij} a_{j,t} \tag{4.39}$$

where C_{ij} is called the damping matrix. The Rayleigh damping model defines the C_{ij} matrix as

$$C_{ij} = \alpha M_{ij} + \beta K_{ij}. \tag{4.40}$$

This choice is motivated by modal analysis, using the fact that K_{ij} and M_{ij} are simultaneously diagonalisable ($M_{ij} a_j = \frac{1}{\omega^2} K_{ij} a_j$). The discretised form of Eqn (4.33) is then transformed into a linear system of uncoupled damped harmonic oscillators. Each mode has the following one-dimensional ODE

$$\ddot{a} + \omega_n \zeta \dot{a} + \omega_n^2 a = 0, \tag{4.41}$$

in the unforced case, where ζ is the damping ratio, and ω is the undamped frequency. Eqn (4.40) is obtained by the back transformation from the ODE to the original form. Thus for a given damping ratio ζ, Eqn (4.40) relates by Eqn (4.41) to frequency ω as follows,

$$\zeta = \frac{\alpha}{2\omega} + \frac{\beta}{2}\omega. \tag{4.42}$$

In order to choose the parameters for a given damping ratio, a commonly used strategy is to select a discrete set of frequencies ω_i and then perform a least square fit to a given damping ratio ζ using Eqn (4.42). Rayleigh damping can also be described as the first two terms in the theory of proportional damping. Higher terms can be added to fit in a larger frequency interval. However, this involves the inverses of the mass and stiffness matrices, restricting the applicability[100] of higher-order damping terms.

4.2.5 Solution procedure for solid $\mathcal{S}$

The procedure to solve Eqn (4.38) is presented, known as the GN22 Newmark algorithm[37] (the external force vector placed on RHS and $\gamma = \beta_1$ and $\beta = 2\beta_2$), the algorithm classifies to the Newmark algorithm family of schemes due to Eqn (4.35) and Eqn (4.36). This is called as the $\mathcal{S}$ solver. The $\mathcal{F}$ term is evaluated by Eqn (4.27), the C_{ij} is computed by RD modelling.

4.3 Relaxation techniques ($\mathcal{R}$)

In this section, the algorithms that solves the root problem $R(x) = 0$ as outlined in the method section is presented, starting from Eqn (2.22) and Eqn (2.23). The algorithms are categorised by the type of relaxation. The Jacobian-Free Krylov Newton methods are techniques[58] that approximate the Jacobian in the Newton iterative methods by differences of the residuals, acceleration of sequence of numbers, generalised to vector fields by the pseudo vector inverse. Pseudo-algorithm schemes are presented for most of the methods. Using no relaxation is called the PICARD method, hence solving Eqn (2.20) iteratively. An iteration is defined as one computation of $\mathcal{S}$ and $\mathcal{F}$. Normally the fluid step is the most expensive; therefore, setting the end condition at the fluid step, the actual number of iterations is one less. The dimensionality is DOFS over interface only (N) and the number of sampled subiterations (I). Usually it is negligible since the system matrix is at most $N \times I$. Original residual techniques formulated these algorithms as NxN but due to instability and large-sized system matrices, these approaches were abandoned very early in the development of the code. The timing for this step is expected to be negligible. The $\leftarrow$ in pseudo-code is a reference to the text for more information.

Algorithm 1: Newmark GN22 integration scheme

if $n = 0$ **then**
  $M_{ij} \leftarrow$ Compute mass Matrix
  $\ddot{a}_i^* = \frac{1}{\beta \Delta t^2} a_i^t + \frac{1}{\beta \Delta t} \dot{a}_i^t + \frac{1-2\beta}{2\beta} \ddot{a}_i^t$
  $\dot{a}_i^* = \frac{\gamma}{\beta \Delta t} a_i^t + \frac{\gamma - \beta}{\beta} \dot{a}_i^t + \Delta t \frac{\gamma - 2\beta}{2\beta} \ddot{a}_i^t$
  $a_i^{t+\Delta t} \leftarrow$ apply predictor, Eqn (2.19).
else
  while $\|r_i\| > \eta$ **do**
    $(K_t)_{ij} \leftarrow$ Compute tangential stiffness, Eqn (4.32)
    $C_{ij} \leftarrow$ Compute damping matrix, Eqn (4.40)
    $(K_t^*)_{ij} = (K_t)_{ij} + \frac{\gamma}{\beta \Delta t} C_{ij} + \frac{1}{\beta \Delta t^2} M_{ij}$
    $r_i = f_i^{e,t+\Delta t} - f_i^{t+\Delta t} - C_{ij} \dot{a}_j^{t+\Delta t} - M_{ij} \ddot{a}_j^{t+\Delta t}$
    Solve $(K_t^*)_{ij} \delta a_j = r_i$
    $a_i^{t+\Delta t} = a_i^{t+\Delta t} + \delta a_i$
    $\dot{a}_i^{t+\Delta t} = \frac{\gamma}{\beta \Delta t} a_i^{t+\Delta t} - \dot{a}_i^*$
    $\ddot{a}_i^{t+\Delta t} = \frac{1}{\beta \Delta t^2} a_i^{t+\Delta t} - \ddot{a}_i^*$
    $f_i^{t+\Delta t} \leftarrow$ Compute internal force
  end while
end if

4.3.1 Jacobi-Free Krylov Newton method

Taking residual of the Eqn (2.22), form the sequence (r_i^k) and the mapping $\tilde{a}_i = \mathcal{H}(a)_i$, then gather the differences $\Delta \tilde{a}_i^{k-l} = \tilde{a}_i^{k-l} - \tilde{a}_i^k$ and $\Delta r_i^{k-l} = r^{k-l} - r^k$,

$$V_{ij}^k = \left[\Delta r_i^0, ..., \Delta r_i^{k-1} \right], \tag{4.43}$$

$$W_{ij}^k = \left[\Delta \tilde{a}_i^0, ..., \Delta \tilde{a}_i^{k-1} \right]. \tag{4.44}$$

After each fixed-point iteration, it is assumed that the Δr_i is a linear combination of the columns of V^k (c_i^k), computed by the LSQR procedure. This computes the interface with inverse Jacobian for Quasi Newton-Raphson using LSQR procedure, hence called as IQN-ILS[76]. The most commonly used procedure to solve the linear system is by the QR factorisation, i.e. solving $R_{ij}^k c_j^k = Q_{ji}^k \Delta r_j$ where Q is orthogonal and R is an upper triangular matrix. By linearity, also

$$\Delta \tilde{a}_i = W_{ij}^k c_j^k \tag{4.45}$$

and from Eqn (2.22) and the minimisation of the residual, $\Delta r_i = \Delta \tilde{a}_i^k - \Delta a_i^k$ where a_i^k is the computed interface vector, i.e. the actual displacement field supplied from

Algorithm 2: IQN-ILS/MIXING

$k = 0$
$\tilde{a}_i^1 = \mathcal{H}(a_i^0)$
$r_i^0 = \tilde{a}_i^1 - a_i^0$
while $\|r^k\| > \eta$ **do**
  if $k == 0$ **then**
    $a_i^{k+1} = a_i^k + \omega r_i^k$
  else
    MIXING : $\Delta\tilde{a}_i^k = \tilde{a}_i^k - \tilde{a}_i^{k-1}$
    IQN-ILS : $\Delta\tilde{a}_i^{k-l} = \tilde{a}_i^{k-l} - \tilde{a}_i^k$
    Construct V_{ij}^k and W_{ij}^k from Eqn (4.43), Eqn (4.44)
    Compute by QR the $V_{ij}^k = Q_{ik}^k R_{kj}^k$
    Solve $R_{ij}^k c_j^k = -Q_{ji}^k r_j^k$
    IQN-ILS : $a_i^{k+1} = a_i^k + W_{ij}^k c_j^k + r_i^k$
    MIXING : $a_i^{k+1} = a_i^k + (V^k + W^k)_{ij} c_j^k + r_i^k$
  end if
  $k = k + 1$
  $\tilde{a}_i^{k+1} = \mathcal{H}(a_i^k)$
  $r_i^k = \tilde{a}_i^{k+1} - a_i^k$
end while

$\mathcal{R}$ to the mesh motion solver $\mathcal{M}$ and $\tilde{a}_i^k$ is the output vector from the solid solver. The update of the interface then becomes

$$a_i^{k+1} = a_i^k + W_{ij}^k c_j^k + r_i^k. \tag{4.46}$$

Another approach is to compute the V^k with respect to the previous iteration, Anderson mixing (MIXING) [59], leading to a slight modification of the scheme. The pseudo code for MIXING and IQN-ILS are collapsed into the same for transparency, see Algorithm 2. Several different approaches to this residual technique. A more direct approach to the minimisation is to apply the method of Lagrange multipliers, requiring an objective function, leading to the Pulay mixing (PULAY) [101] see Algorithm 3,

$$a_i^{k+1} = c_l a_i^l, \tag{4.47}$$

where c_l is the Lagrange multipliers and $0 \leq l \leq k$. The weight (c_l) is computed from applying the objective function $L = \|c_l r^l\|^2 - 2\lambda(c_l 1_l - 1) = c_k B_{kl} c_l - 2\lambda(c_l 1_l - 1)$, where $B_{kl} = r_i^k r_i^l$.

Algorithm 3: PULAY

$k = 0$
$\tilde{a}_i^1 = \mathcal{H}(a_i^0)$
$r_i^0 = \tilde{a}_i^1 - a_i^0$
while $\|r_i^k\| > \eta$ **do**
  Construct X_{ij}^k
  Solve Eqn (4.48)
  $a_i^{k+1} = c_l a_i^l$
  $k = k + 1$
  $\tilde{a}_i^{k+1} = \mathcal{H}(a_i^k)$
  $r_i^k = \tilde{a}_i^{k+1} - a_i^k$
end while

A direct computation of the derivatives and setting to zero leads to the following

$$X_{ij}^k c_j = \begin{bmatrix} B_{00} & B_{01} & \dots & B_{0k} & -1 \\ B_{10} & B_{11} & \dots & B_{1k} & -1 \\ B_{20} & B_{21} & \dots & B_{2k} & -1 \\ B_{30} & B_{31} & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & B_{kk} & -1 \\ -1 & -1 & \dots & -1 & 0 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ \vdots \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ -1 \end{bmatrix}. \tag{4.48}$$

A related technique is the GMRES applied to the interface vector, so-called IGMRES [102], which applies Krylov technique upon the residual using orthogonalisation, see Algorithm 4. The IGMRES, IQN-ILS, MIXING and PULAY form the Jacobi-Free Krylov Newton methods (JFKN). All these methods are equivalent to each other for linear $\mathcal{H}$. Normally, one applies an upper limit to the number of previous iterations (taps) to be used, which, of course, affects the performance; therefore, this number is within the parenthesis of the method name.

4.3.2 Acceleration of sequences

Acceleration techniques, such as Aitkens δ^2 (AITKEN) [57] and successive under-relaxation (SUR), belong to a category of relaxation phrased as BLEND. The essential relaxation in BLEND is with a given $\omega \in (0, 1]$

$$a_i^{k+1} = a_i^k + \omega r_i^k. \tag{4.49}$$

In addition to BLEND, there is a hybrid between BLEND and JFKN, namely the DSUR technique, which computes the optimal blend ω^k from the previous iterations

Algorithm 4: IGMRES

$k = 0$
$\tilde{a}_i^* = \mathcal{H}(a_i^*)$
$r_i^* = \tilde{a}_i^* - a_i^*$
while $\|r_i^k\| > \eta_1$ **do**
  $j = 0$
  $\xi = \|r_i^k\|$
  while $\xi > \eta_2$ **do**
    $j = j + 1$
    for $l = 1$ to $l = j - 1$ **do**
      $\Delta a_i^j = \Delta a_i^j - \frac{\Delta a_m^j \Delta a_m^l}{\|\Delta a_i^l\|} \Delta a_i^l$
    end for
    $\Delta a_i^j = \frac{\|a^*\|}{\|\Delta a_i^j\|} \Delta a_i^j$
    $a_i^j = a_i^* + \Delta a_i^j$
    $\tilde{a}_i^j = \mathcal{H}(a_i^j)$
    $V_{ij}^j = [\Delta r_i^1 .. \Delta r_i^j]$
    $W_{ij}^j = [\Delta a_i^1 ... \Delta a_i^j]$
    Compute QR, $V_{ij}^j = Q_{ik}^j R_{kj}^j$
    Solve $R_{il}^j c_l = -Q_{li}^j r_l^k$
    $\xi = \|r_i^k + V_{li}^j c_l\|$
  end while
  $a_i^* = a_i^* + W_{il}^j c_l$
  $r_i^{k+1} = \mathcal{H}(a_i^*) - a_i^*$
  $k = k + 1$
end while

by LSQR and computes the average $\langle \omega^k \rangle$ within a moving time window, where the width of the window is set as the maximum number of subcycles,

$$\tilde{\epsilon}^k = min_{\omega^k \in [0,1]} \|a_i^{n-1} - (a_i^k + \omega^k r_i^k)\|. \tag{4.50}$$

It is applied by comparing the residual value of the current step with the residual from the previous, and if smaller, then apply the current estimate; otherwise, apply the average value.

$$a_i^* = \begin{cases} a_i^k + \omega^k r_i^k, \ \tilde{\epsilon}^k < \|r_i^k\| \\ a_i^k + \langle \omega^k \rangle r_i^k, \ \tilde{\epsilon}^k \geq \|r_i^k\| \end{cases} . \tag{4.51}$$

See algorithm 5 for an outline of the methods. Of course, JFKN restricted to two vectors is a special case of BLEND.

Algorithm 5: BLEND

$k = 0$
$\tilde{a}_i^0 = \mathcal{H}(a_i^0)$
$r_i^0 = \tilde{a}_i^0 - a_i^0$
while $\|r^k\| > \eta$ **do**
  AITKEN:
  if $k == 0$ **then**
    $\omega^0 = sign(\omega^n) min(|\omega^n|, \omega^{max})$
  else
    $\omega^k = -\omega^{k-1} \frac{r_i^{k-1}(r_i^k - r_i^{k-1})}{(r_i^k - r_i^{k-1})(r_i^k - r_i^{k-1})}$
  end if
  SUR: ω^k fixed value
  DSUR: Compute ω^k from Eqn (4.50) and Eqn (4.51)

  $a_i^{k+1} = a_i^k + \omega^k r_i^k$
  $\tilde{a}_i^{k+1} = \mathcal{H}(a_i^k)$
  $r_i^{k+1} = \tilde{a}_i^{k+1} - a_i^{k+1}$
end while

4.3.3 Polynomial extrapolation

Another type of acceleration is extrapolation methods[24]: the minimal polynomial extrapolation (MPE) and Reduced Rank Extrapolation (RRE). They are related to Krylov subspace techniques, arising in the context of full orthogonalisation method and generalised minimal residuals, Krylov technique that applies to solve linear systems $(\delta_{ij} - T_{ij})x_j = d_i$. Defining the residual $r_i^k = a_i^{k+1} - a_i^k$, with the fixed-point equation Eqn (2.20) to be solved, given a sequence of successive approximations (a_i^k), acceleration of sequence, as with blending, find an approximation sequence of the form,

$$s_i^k = \gamma_l a_i^l = a_i^0 + \xi_l r_i^l, \tag{4.52}$$

where $\gamma_l 1_l = 1$ and $\xi_0 = 1 - \gamma_0$ and $\xi_l = \xi_{l-1} - \gamma_l$ for $0 \leq l \leq k$. Then $s = lim_{k\to\infty} s_k$ is the solution to the fixed-point solution. This can also be expressed as

$$s_i^k = a_0 + U_{ij}^{k-1} \xi_j, \tag{4.53}$$

where $U_{ij}^{k-1} = [r_i^0 ... r_i^{k-1}]$. Now for MPE, one computes the γ_i by the following,

$$min_{c_l} \|c_l r_i^l + r_i^k\|, \tag{4.54}$$

Algorithm 6: MPE/RRE

Compute $r_i^l = a_i^{l+1} - a_i^l$ for $0 \leq l \leq k$
Compute $U_{ij} = [r_i^0 ... r_i^k]$
Compute QR of U_{ij}, take subset Q_{ij}^{k-1}, R_{ij}^{k-1}.
MPE:
Solve $R_{ij}^{k-1} c_j = -Q_{ij}^{k-1} r_j^k$
$c_k = 1$ and $\gamma_l = \frac{c_l}{c_m 1_m}$
RRE:
Solve $R_{mi}^k R_{mj}^k c_j = 1_i$
$\gamma_l = \frac{c_l}{c_m 1_m}$
Compute for both MPE/RRE:
$\xi_0 = 1 - \gamma_0$ and $\xi_l = \xi_{l-1} - \gamma_l$ for $1 \leq l \leq k-1$
$s_i^k = a_i^0 + Q_{im}^{k-1} R_{ml}^{k-1} \xi_l$

and then $\gamma_l = \frac{c_l}{c_m 1_m}$ where $0 \leq l \leq k-1$ and $c_k = 1$. On the other hand, RRE computes γ_l by the following,

$$min_{\gamma_l} \|\gamma_l r_i^l\|, \tag{4.55}$$

with the condition $\gamma_m 1_m = 1$. And the solution then becomes $s_i^k = \gamma_m a_i^m$, or expressed by Eqn (4.53). See algorithm 5 for an outline of the implementation.

4.3.4 Epsilon algorithm

The epsilon algorithm also known as Wynn's method [103] is an acceleration scheme for computation limits of sequence of scalars (s_i). It should be emphasised the strong correlation between Aitkens δ^2, MPE, and RRE with this method; for a more detailed discussion, see the work of Sadok et al [104]. The method is straightforwardly described by the following scheme, the epsilon table,

$$\begin{array}{ccccc} & s_0 & & & \\ 0 & & \epsilon_1^{(0)} & & \\ & s_1 & & \epsilon_2^{(0)} & \\ 0 & & \epsilon_1^{(1)} & & \ddots \\ & s_2 & & \epsilon_2^{(1)} & \\ 0 & & \epsilon_1^{(2)} & & \\ & s_3 & & \epsilon_2^{(2)} & \\ \vdots & & \vdots & & \vdots \end{array} \tag{4.56}$$

The leftmost column and second column are the initialisation, $\epsilon_{-1}^{j} = 0$, $\epsilon_{0}^{j} = s_j$. The iteration is

$$\epsilon_{k+1}^{(j)} = \epsilon_{k-1}^{(j+1)} + [\epsilon_{k}^{(j+1)} - e_{k}^{(j)}]^{-1}. \tag{4.57}$$

The sought answer is the right-most middle row value of $\epsilon_n^{n/2}$. This is generalised to vector sequences by pseudo inverse $a_i^{-1} = \frac{a_i}{\|a_i\|^2}$. This is denoted as the vector-epsilon algorithm (VEA), see Algorithm 7. The algorithm uses matrices ep, ec, en to represent the terms in Eqn (4.57), where $ec = \epsilon_k^j$ and $ep = \epsilon_{k-1}^{j+1}$ and $en = \epsilon_{k+1}^j$ and in update to next iteration $ep = ec$ and $ec = en$. The ec is the notation for the current epsilon, i.e. ϵ_k^*, and likewise ep for the previous epsilon (ϵ_{k-1}^*) and finally next epsilon (ϵ_{k+1}^*). The k index refers to column and j the row index of the scheme (4.56).

Algorithm 7: VEA

ec=[a_i^0 ... a_i^k],ep=0
n=0;
while column size of ec>1 **do**
  n=n+1
  en=[0_i ... 0_i] , column size=column size of ec-1
  for j<column size of en **do**
    $en_{ij} = ep_{i(j+1)}$
    $v_i = ec_{i(j+1)}$
    $v_i = v_i - ec_{ij}$
    $v_i = v_i / v_k v_k$
    $en_{ij} = en_{ij} + v_i$
  end for
  ep=ec
  ec=en
end while
if $n\%2 == 0$ **then**
  a_i^* = ec_i
else
  a_i^*=last column of ep_{ij}
end if

A variant of VEA is the so-called topological-epsilon algorithm (TEA) expressed in vector form,

$$(\epsilon_{2k+1}^{(j)})_i = (\epsilon_{2k-1}^{(j+1)})_i + \frac{y_i}{y_l(\Delta\epsilon_{2k}^{(n)})_l}, \tag{4.58}$$

$$(\epsilon_{2k+2}^{(j)})_i = (\epsilon_{2k}^{(j+1)})_i + \frac{(\Delta\epsilon^{(j)})_i}{(\Delta\epsilon_{2k+1}^{(j)})_l(\Delta\epsilon_{2k}^{(n)})_l}. \tag{4.59}$$

where y_i is orthogonal vector to the residuals of s_i, and Δ forward difference operator, see Algorithm 8.

Algorithm 8: TEA

ec=$[a_i^0$ ... $a_i^k]$,ep=0
while column size of ec>2 **do**
  en=$[0_i$ ... $0_i]$, column size=column size of ec-1
  for j<column size of en_{ij} **do**
    en_{ij}=$ep_{i(j+1)}$
    v_i=$ec_{i(j+1)}$
    v_i=$v_i - ec_{ij}$
    en_{ij}=$en_{ij} + \frac{y_i}{y_k v_k}$
  end for
  enp_{ij}=$[0_i$ ... $0_i]$, column size=column size of en-1
  for j< column size of enp_{ij} **do**
    enp_{ij}=$ec_{i(j+1)}$
    v_i=$ec_{i(j+1)}$
    v_i=$v_i - ec_{ij}$
    $w_i = en_{i(j+1)}$
    $w_i = w_i - en_{ij}$
    enp_{ij}=$enp_{ij} + \frac{v_i}{w_k v_k}$
  end for
  ep=en
  ec=enp
end while
a_i^*=last column of ec_{ij}

4.3.5 Broyden algorithm

The approach applied here is called the "bad" Broyden's method[105], and it is a secant method. The idea is to update the Jacobian iteratively by rank one matrix and by

linearity and defined multiplication for JFKN, the update formula becomes compact. The algorithm can be summarised as for $r_i^j = \tilde{a}_i^j - a_i^j$ and

$$z_i = z_i + r_i^{j+1} \frac{r_l^j z_l}{\|r_i^j\|^2}, 0 \leq j \leq k-1, \tag{4.60}$$

$$r_i^{k+1} = \frac{z_i}{1 - \frac{r_l^k z_l}{\|r_i^k\|}}, \tag{4.61}$$

$$a_i^{k+1} = a_i^k + r_i^{k+1}. \tag{4.62}$$

See Algorithm 9 for the procedure.

Algorithm 9: BROYDEN

z_i=-r_i^k
for j<k **do**
  z_i=$z_i + r_i^{j+1} \frac{r_l^j z_l}{r_l^j r_l^j}$
end for
$r_i^{k+1} = -\frac{z_i}{1 - \frac{r_l^k z_l}{\|r_i^k\|}}$
$a_i^* = a_i^k + r_i^{k+1}$

4.3.6 Hybrid techniques

The relaxation step, similar to the action steps, can be applied to both input and output. Further, it is expected that the efficiency of each algorithm may vary by different dependencies. In combining different relaxations and/or applying them at different steps, tokenisation is applied by setting a token for each method, switch token and number at which iterative step a method is to be applied, see Table 4.1 or within parenthesis. For different steps (-), normally it applies to solid/fluid. The switch token says that only the left/right (< or >) side of the string is applied if lesser/greater than the mentioned number; any tokens placed between split token is applied for all coupling iterations. The split rule k, is a switch where all blending techniques are applied below or equal k and residual techniques above. As implied, some methods are more efficient towards the end of the coupling or at the beginning. Several different strategies have been evaluated, and only those deemed prominent are presented in paper IV. One of particular interest is the bombardment technique[106] in which one computes the residual for a set of relaxations, and then as a selection principle, one can either take the lowest norm (T), make an LSQR approach (Q), or do the average (default).

Table 4.1: Tokenisation for BOMBARDMENT with split token

Token	Relaxation
I	IQN-ILS
D	DSUR
P	PULAY
A	AITKEN
X	MIXING
S	SUR
<,>	switch tokens

Chapter 5

Implementation and validation

This chapter presents the implementation details and steps of validation of the FSI code. It begins with a presentation of the used software packages, then for each package, snippet codes and code strategy is presented, in such way that the FSI code can be reconstructued, using chapter 4. Then the code strategy of the MPI implementation is presented, how to measure the scaling. The validation procedure marks the final part of the chapter and presenting the main steps in providing verification and validation. Mostly it is done by first verifying each solving step on a stand-alone unit test, then the results of common benchmark cases are presented, which validates the FSI solver.

5.1 Software packages

The fluid and solid steps are built on the OpenFOAM[107] and deal.II[62] packages, respectively. The interface is an in-house solution. The OpenMPI package[27] is used for parallellisation. The partition manager used for creating partition for the MPI, deal.II employs while OpenFoam uses scotch. Similarly, numerical libraries (OpenBLAS, LAPACK, BLAS, etc.) are involved in both packages by dependency. However, deal.II also employs PetSc, a scientific library for solving parallel PDE problems. The FSI code is adapted to Paraview for visualisation and postprocessing. The preprocessing is mostly centred around mesh generation, using the GMSH package[108], a stand-alone program using script input files and generates an unstructured mesh for the fluid and structured mesh for the solid. Matlab/Octave is used for postprocessing together with scripting to be used for gnuplot and the graphs. After the implementation and validation of the FSI core solver, the versions of the packages were not upgraded with later releases, in order to avoid biased results, so the used versions are OF2.3.X and deal.II 8.0.

5.2 Implementation

The description of the implementation is split into sections, each explaining: the details of the interface code, which is the main program of the FSI solver, and then each state solver, the parallellisation approach, and relaxation procedure. The Open-

```
Load Mesh
Mapping between fluid and solid
Loop over Time
    while End condition is not satisfied
       for each token in tokenisation string
         switch (token)
            case 'F':
           // code for executing Fluid step solver
           .... etc ...
    saving data for postprocessing
```

Figure 5.1: Pseudo code of the main code using tokenisation, see Table 5.1 for available tokens.

FOAM provides a state solver for fluid, which is code-wise split into an INS solver and a mesh motion solver. The deal.II package provides a toolbox for the parallelisation, construction of the shape functions, assemblage, and encapsulated PetSc to solve a system of linear equations. Using the namespace programming technique, the governing variables of the coupled interface are stored at run time in RAM memory, allowing transferring data between state solvers without using any file system. The tokenisation procedure implemented in the main program accesses each solver, driven by a stream of tokens, provided as a string from input. Each state solver has an input file, describing the conditions, material properties, and mesh.

5.2.1 Interface code

The interface code contains the tokenizer, the transfer functionality between the state solvers, acting as a dereference pointer or namespace variable depending on which variable and which step. It is the main program, that integrates the OpenFOAM and deal.II into one executable code.

Flowchart: main code

The FSI procedure, see Figure 5.1, begins with reading the case description and input files. It then computes the mapping functions and provides the largest distance between mapped facets cell centres, the settings are such that the facet centre of the fluid is onto the solid facet. The rest of the main routine is a nested time loop processing the token string. The token string defines the steps to be taken during one subiteration. It shall always for FSI contain a fluid, solid and motion step. Then for fluid only, the solid is omitted and correspondingly for solid only, the fluid step is omit-

ted. Normally there is a default setting, the "!SMF" is the default string, see Table 5.1 for the most commonly applied tokens. Another advantage of tokenisation is that programmatic instruction can be applied, allowing for example different strings to be applied, if divergence or when any other exceptions occurs defined by conditions. Since action steps can be between steps or within steps, they are specified elsewhere in the input.

Table 5.1: Tokenisation for partition solver.

Token	Instruction
F	INS solver
M	ALE mesh solver
S	Solid solver
!	MPI Synchronisation
R	Relaxation
@	End condition

Action steps

All action steps defined in the study are implemented in the interface and thus applied only to the variables stored within that module, i.e. the field variables of the coupled boundary. The following shows at which transition the action applies,

- $\mathcal{S} \rightarrow \mathcal{M}$: filtering, relaxation, inverse interpolation, extrapolation
- $\mathcal{F} \rightarrow \mathcal{S}$: filtering, relaxation, extrapolation
- $\Delta t \rightarrow \Delta t + 1$: extrapolation
- $\mathcal{M} \rightarrow \mathcal{F}$: filtering, relaxation, extrapolation

Finite Impulse Response (FIR) filtering is applied to suppress noisy data. Let the $x[i]$ be defined as the incoming signal at observation step i, which can be a macro time step or sub-iteration index. Assume n such steps have been sampled (i.e. taps), then the outgoing signal step x[n] can be expressed as,

$$x[n] = \sum_{i=0}^{m-1} b_i x[n-i], \tag{5.1}$$

where $m < n$ is the filtering order, and the expression of b_i depends on the filter window, which for the current study is either a constant value or the Blackman window. The sum $\sum b_i$ should be 1 in order not to alter a constant field. For white noise error,

m=1, and b_i=0.5, e.g. the average, is the most efficient choice. Using FIR degrades the order of accuracy of the procedure to the first order, but this is negligible for sufficient large sampling over the period of interest. This is normally applied in this study as pre-processing, and if so, it will be phrased as FIR followed by a cutoff frequency and type of filter. To transfer data across a map that is non-conforming with no common

```
for (int i=0; i<FOAM2DEAL.size();i++) {
     if (FOAM2DEAL.at(i)== solid_index) {
       Point<dim> center=FLUID.at(i);
       // interpolated quadrature point (quadidw)
       double w =center.distance(quadidw)+SMALL;
       P+=TRACTION.at(i).at(6)*std::exp(nweight*std::log(w));
       // CAUCHY VISCOUS STRESS
       double f=std::exp(idwexpf*std::log(w))*J;int kl;
       for (int k=0;k<dim;k++) for (int l=0;k<dim;l++) {
        kl=k>l?(l*dim+k-(l*(l+1))/2): k*dim+l-(k*(k+1))/2;
        C(k,l)=f*TRACTION.at(i).at(kl);
       // Nominal Force : JFiC
       Fi.mTmult(S,C,true);
       weight+=std::exp(idwexpf*std::log(w));
    }
}
// project along the normal (sigma*n)
S.vmult(t,n,false);
traction.at(0)=t(0);traction.at(1)=t(1);traction.at(2)=t(2);
traction.at(3)=P;
// ending function call
return t/=weight;
```

Figure 5.2: Code snippet, the traction split for Eqn (5.6), the traction contains the stress for viscous traction and the pressure. The traction usage is part of the function call $get\_traction$ which has the input parameter $solid\_index$, interface code provides the inclusion maps, computes the input reference quadrature point to current, then viscous stress is scaled according to IDW formulae. After looping, the traction is computed and scaled by the total weight and this finalise the IDW computation of the traction.

nodal points, only a common interface surface, requires an interpolation procedure to determine the function value f at a given target point x_i on the coupled interface Γ at the current configuration. For each target point, x_i there will be a set of source points x_i^n on the coupled surface, which are part of the adjacent domains mesh with corresponding function values $f^*(x_i^n)$. See Figure 5.2. The interfaced mapping stores for the deal.II four displacement points/vertex per solid facet and vertex point of the fluid facet, then the scoring index for the closest vertex to the given quadrature point or vice versa, sets the data for the IDW interpolation in both directions.

Let the weight w_n between a target point and a source point be defined as

$$w_n = \frac{1}{\|x_i - x_i^n\|^d + \epsilon}, \tag{5.2}$$

where ϵ is arbitrarily small but positive. These weights are computed at every mesh update. These weights are then applied to the inverse distance weighting formulae (IDW)[70] defined by

$$f(x_i) = \frac{1}{w_k 1_k} w_n f^*(x_i^n). \tag{5.3}$$

As mentioned, since the interpolation is executed at the current configuration, the solid weights are based upon the transformed solid facets, obtained by summation of reference configuration and displacement field.

One can further stabilise the procedure by preconditioning the input variables. Even though the following discussion can be applied to any field, assuming that for a given displacement field a_i^n the corresponding traction $t_i^n = \mathcal{F} \circ \mathcal{M}(a_i^n)$ is given, where superindex n represents step n, which is either a time step index or sub-iteration index. The common goal for all loading and traction control is to incrementally change the field value from a reference value to its natural value by using a ramp function so that one may control the acceleration of the structure and thus allow building up the inertia in a stable way. Then given the computed traction t_i for given step n, a reference traction (t^*) and a blending s^n, an intermediate value t_i^n is applied instead of the computed value,

$$t_i^n = s^n t_i^* + (1 - s^n) t_i, \tag{5.4}$$

where $s^n = \phi(n)\psi$ with $0 < \phi < 1$ and $0 < \psi \leq 1$. The blending function ϕ gradually goes to zero monotonically as n increases, and thus the intermediate value t_i^n goes to the computed value t_i. This has been applied to balance equation as well, then known as the continuation technique, allowing precondition or acting as relaxation.

Before the non-linear iterative steps in the finite strain algorithms, a predictor (a^*) of the displacement was applied, see Eqn (2.19), using first and second-order terms in the Taylor expansion over time. Another approach is, by finite difference, to extrapolate the current state from previous states, either used as a replacement of the state solver or trial vector. Of course, for noisy data, polynomial fitting by LSQR and then extrapolation is normally applied for the pressure field.

Traction split

The second Piola–Kirchhoff stress tensor (S_{ij}) applies to the reference configuration, while the Cauchy stress tensor (σ_{ij}) to the current configuration. The synchronisation

provides the latter from the fluid step as input for solid step, and thus requires a transformation, using the so-called Nanson formulae,

$$S_{ij} = JF^{-1}_{ik}F^{-1}_{jm}\sigma_{km}, \tag{5.5}$$

which relates to the nominal traction (t_0) Eqn (4.28) by the first Piola-Kirchhoff tensor $N_{ij} = S_{ik}F_{jk}$ and $(t_0)_j = N_{ij}(n_0)_j$, where $(n_0)_i$ is the normal in reference domain. Therefore, code-wise, the transfer is done with the first Piola-Kirchhoff since the return statement is only the traction t_0. The traction on the fluid side is split into two components: the viscous stress, computed by the symmetric gradient of the velocity ($\sigma^{visc}_{ij} = \frac{\nu}{2}(U_{i,j} + U_{j,i})$), and the pressure ($p^f$). The nominal force at the interface is then computed as follows, assuming matched fluid facet onto solid facet,

$$(t_0)_i = JF^{-1}_{ik}\sigma^{visc}_{kj}(n_0)_j - p^f(n_0)_i, \tag{5.6}$$

this is the traction split. However, it's not exact since the fluid facet is not exactly matched onto the solid facet; although its centre is on the solid facet, the normal wiggles, and with increasing resolution it aligns with the opposite normal. Further, since there is a possibility of multiple fluid cells with the same solid cell as the closest facet, the IDW formula is applied for the mapping of traction at the quadrature point of the solid. As a remark: The difference in use of the Piola-Kirchhoff tensors is the involvement of the Green strain tensor or not in the term of interest, on the RHS of Eqn (4.32) there is no E_{ij} terms on the external force term, the traction, but the internal force tensor has.

5.2.2 OpenFOAM package

The OpenFOAM C++ tool kit is exemplified by computing the moment predictor step Eqn (4.9), which is evaluated using class $fvVectorMatrix$. This class discretises and distributes the operators specified within curled brackets, the code extracted from include file UEq.H is shown in Figure 5.3. It is used in the solver that solves the INS with moving patches, pimpleDyMFoam. The class first applies the constructor using fvm class which is the solution method applied, namely FVM. To discretisation, in this case, $fvm :: ddt(U)$ contains the scheme for the temporal part, $fvm :: div(phi, U)$ refers to the scheme for the divergence operator with involving variables phi and U, and the two remaining terms are the effective viscosity part as mentioned Eqn (4.5), the $\nu_{eff}(-\frac{2}{3}U_{k,k}\delta_{i,j} + U_{i,j} + U_{j,i})$. Some snippets from the PISO algorithm are presented: from $UEqn$ class one gets the a_P terms from the method call $A()$, from which the intermediate velocity is computed, then the pressure equation can be formulated and solved, by using the computed flux, the pressure correction to the velocity see Figure 5.2. The Fvc class refers to the Finite Volume Calculus class, which contains the interpolation schemes.

```
fvVectorMatrix UEqn
{
    fvm::ddt(U)
   +fvm::div(phi,U)
   -fvm::laplacian(nuEff(), U)
   -fvc::div(nuEff()*dev(fvc::grad(U)().T()))
}
solve(UEqn==-fvc::grad(p));
```

Figure 5.3: Code snippet, the moment predictor: $A_{ij}\phi_j = R_i$.

Table 5.2: Code snippets used in OpenFOAM.

Eqn (4.6)	a_P	AU=UEqn().A()
Eqn (4.12)	$U_i = U_i - \frac{1}{a_p} p_{,i}$	U-=fvc::grad(p)/AU
intermediate velocity	$U_i^* = \frac{1}{a_P} H[U_j]_i$	U=UEqn().H()/AU
volumetric flux	$\phi^* = S_k U_k^*$	phi=fvc::interpolate(U)& mesh.Sf()
Eqn (4.13)	$(\frac{1}{a_P} p_{,i}))_{,i} = 1_i \phi_{,i}^*$	fvm::laplacian(1.0/AU, p)==fvc::div(phi)

5.2.3 deal.II package

The core deal.II functionality in computing the integral values is exemplified by the following snippet code of the evaluation of the surface integral of traction vector. The

Table 5.3: Correspondence table between notation and variables in deal.II. w_i is the quadrature weight.

K_t	system_matrix
f	system_rhs
$H_{iq}(\xi_q)$	fe.values.shape value(i,q)
ξ_q	fe.values.quadrature point(q)
$\|detJ(\xi_q)\|w_q$	fe values.JxW(q)

deal.II functionality is first exemplified by the following snippet code of the evaluation of the surface integral of traction vector, see Table 5.3, and computes traction term of the external force over one solid facet approximated by Gauss quadrature integration formula,

$$\int_{\partial CV} \frac{1}{\rho_s} H_{jk} t_j dA \approx \sum_q \frac{1}{\rho_s} H_{jk}(\xi_q) t_j(\xi_q) |detJ(\xi_q) w_q. \tag{5.7}$$

The snippet code for this in deal.II is presented in Figure 5.4. The first loop extends over the faces of a given cell. The boundary indicator() ==1, ensures the surface face is chosen, and the $reinit$ member function initialises the values of all test functions related to a given cell. The inner loop is over the quadrature points. The second example

```
double invrho=1/rho,load;
const unsigned int n_face_q = face_quadrature_formula.size();
int comp_i = fe.system_to_component_index(i).first;
for (unsigned int face=0;
face<GeometryInfo<dim>::faces_per_cell;++face)
if ( cell->face(face)->boundary_indicator() == 1 ) {
   index=cell->face_index(face);
   fe_face_values.reinit(cell,face);
   for ( unsigned int q_point=0;q_point<n_face_q;++q_point){
     // traction split
     traction=shared::icofsi_db_list[n_object]->get_traction
     (
      cell->face_index(face),
      a,
      fe_face_l_values.quadrature_point(q_point),
      fe_face_l_values.normal_vector(q_point)
     );
     double nj;
     nj=fe_face_l_values.normal_vector(q_point)(component_i);
     load=traction(component_i)-traction(3)*nj;
     load*=invrho;
     double weight=fe_face_l_values.JxW(q_point);
     double face_val=fe_face_l_values.shape_value(i,q_point);
     cell_rhs(i)+=load*face_val*weight;
  }
}
```

Figure 5.4: Code snippet, for $(t_0)_j$ in Eqn (5.7).The $boundary\_indicator()$ classify the patch, "1" imply coupled FSI patch. The $get\_traction$ is the interface matchmaker, with the input of the face index, quadrature point and normal vector, it computes with IDW at current configuration the traction for given quadrature point, then apply pullback, obtaining the $traction$ for reference. It comes in two components, firstly the viscous term (σ*normal), secondly the pressure, which is the essence of the traction split. The innerloop is over quadrature points, the face integral is computed with the quadrature weight as outlined in Table 5.3.

is to show the crucial difference between OpenFOAM and deal.II. The OpenFOAM benefits from moving the discretisation into classes and allowing abstract treatment of PDE by all the tricks in the book of C++ functionality. It significantly reduces the effort in the implementation of solvers. In deal.II version used, there is no such solver except for limited tutorials, the code snippet illustrates the assembly of the system matrix, before it calls the solver of the linear equation.

Instead of allowing polymorphism to '=*+/-' math operators, and having a class that deals with the discretisation, deal.II deals with methods for each operation such as add, $copy\_from$, $assign$, $vmult$, $mmult$ and so forth.

```
// current subiteration k with maximum number of iteration I
int I,k;
// inside subiteration loop
// update bundle_di=SoF(bundle_xi)
trim_bundle(bundle_di,k);
update_bundle(bundle_di,tilde_u_cur,k);
//update bundle_ri for k
trim_bundle(bundle_ri,k);
update_bundle(bundle_ri,bundle_di,k,k,0);
update_bundle(bundle_ri,bundle_xi,k,k,-1.0);
if (iqnils_initialised ) {
// X number of taps
    create_delta(delta_di,bundle_di,std::min(X-1,k));
    create_delta(delta_ri,bundle_ri,std::min(X-1,k));
    // QR decomposition
    std::vector<double> ci=handler.qr_house_solve(
         delta_ri,
         bundle_ri.at(k%I) );
    // solve IQN-ILS assertion a=ai+ri+solution*ci
    std::vector<double> solution=handler.vmult(
        delta_di,
        ci);
    // forming the solution vector
    vector_add(solution,bundle_ri.at(k%I),1.0);
    vector_add(solution,bundle_xi.at(k%I),1.0);
    // set_displacement transfer data to the MPI transferring
    solution_mpi=0; solution_mpi=boundary_field(solution_mpi,solution);
    // transferring to the global vector
    displacement_field(solution_mpi); solution.resize(0);
  }
  handler.clear(delta_ri);handler.clear(delta_di);
```

Figure 5.5: Code snippet, IQN-ILS procedure presented in Algorithm 2 in chapter 4 using economic QR procedure. The interface CMatrixTool is used as a "handler" class, that is a class that takes input variables, executes desired operation, and returns the results. $trim\_bundle$ and $update\_bundle$ handles the stack of global vectors, and the computing of difference matrices is done by $create\_delta$. QR is solved by qr_house_solve equation and solution computed by $vmult/vector\_add$. The solution in global vector is inserted into the interface on the mpi solution vector, then it replaces the current interface on global stack by $displacement\_field$. Garbage collection is handled by $clear/resize(0)$

5.2.4 Relaxation step

The relaxation uses the CMatrixTool, which is an in-house math library using C++ Standard Template Library (STL) functions. This is exemplified by showing how the IQN-ILS is implemented, see Figure 5.5. The CMatrixTool uses the same method names as deal.II for keeping transparency within the code. The handler is a class that takes the in parameter, then performs the math procedure, keeping necessary data

encapsulated, then allowing further processing, and also acting as a garbage collector. The advantages of STL are many, apart from being efficiently implemented, it also ensures data integrity. The word bundle refers to a stored collection of previous solution vectors (X), and the routine begins with adding the latest solution from solid step, i.e. $x^{k+1} = \mathcal{H}(x^k)$, the $tilde\_u\_cur$ vector, to this bundle by first trimming the bundle $trim\_bundle$ which extracts the coupled boundary and storing it, by merely handle a dynamic list stack, LIFO type, using $push\_back$, a pointer $k\%I$ gives the position of the current vector. Then it adds to the bundle the latest solution and updates the residual by $update\_bundle$. The W_{ij}^k and V_{ij}^k difference matrices are created by $create\_delta$ and at the earliest second subiteration the IQN-ILS procedure begins by solving the residual equation by using the QR method in the handler, $handler.qr\_house\_solve$, several different QR procedures are implemented, the one used here is the economic QR, using Householder QR, which eliminate the explicit usage of Q matrix, otherwise the memory usage will be too large (NxN). The $vector\_add$ is a routine that handles the extraction of the data from the bundle into a solution vector, and then one changes the format, from STL to deal.II, for the purpose of updating the global interface vector. The logistic difficulty, to know which process the relevant DOF belongs to, and the relation between the location of interface to the position in the solution vector, can be simplified by consistently transferring data in the same order that they are processed in solid step. This also minimises the bandwidth. The in/out of data thus follows the same processing path, no need for keeping track of the data: the $displacement\_field$ transfer and mpi solution vector to global vector, the $boundary\_field$ replaces the interface of a mpi vector with the values of the global vector. All remaining algorithms in the relaxation step, use the same functionality as outlined here. The translation of the pseudo code into C++ is almost seamless by this approach.

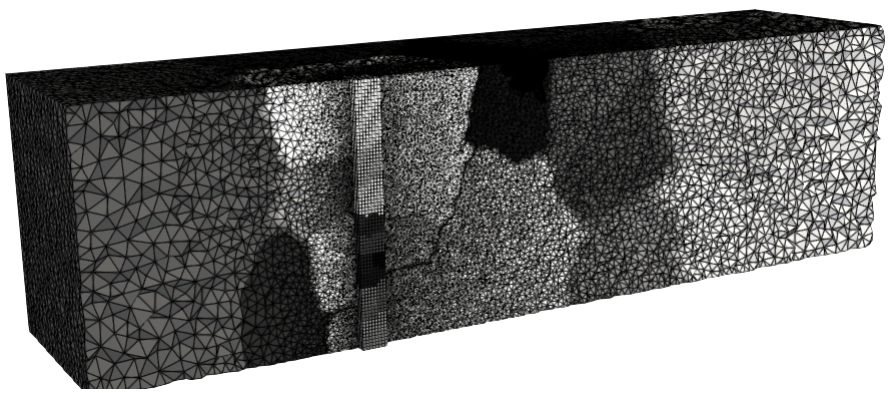

Figure 5.6: MPI decomposition. Cell extract from mid-cut plane for a channel with solid clamped-clamped cantilever, and each partition is shaded with discrete value, one partition per thread. This mesh is partitioned for two node simulation (40 processes)

5.2.5 OpenMPI: parallelisation

There are several other packages that provides same functionality as described in this section, for example MPICH, the essential difference between those, is that OpenMPI is more modular and easier to implement, while the latter is more efficient. By parallelisation, the field variables of the mesh are distributed across a cluster of processes, creating partitions, assigned to each process, which have disjoint memory and often across different machines, unless within the same computer node. The cells are coloured by which partition they belong; see Figure 5.6 as an example, with a clamped-clamped cantilever immersed in a channel flow. For OpenFOAM, the scotch package is employed for the purpose of creating this partition, while parMetis is applied for deal.II. The cells that share facets with adjacent patches create a ghost-interface, included in each partition for the closure of the solution matrix for the state solvers. Although the way the parallelisation is handled is different between the interface/OpenFOAM and deal.II, the basic strategy is the same since the same communication protocol is applied, the MPI protocol.

MPI protocol

The code is the same across all processes, the code parts that are subjected to parallelisation begin with a marker, and they end with a corresponding end marker, normally a function call. The process ID is used in a workload partition, e.g. to establish loop limits so each process accesses only the relevant data for its process. Communication is by gathering / scattering / synchronisation / wait calls within the MPI package. A normal strategy is to gather all data relevant to each process, execute the code with the acquired data, and then to distribute the results to relevant processes. The wait call is to ensure that all processes have finished before applying synchronisation.

Efficiency loss

Efficiency loss occurs when each process must wait until all processes reach the end marker, and may occur either by different partition sizes or workload, either by own code or competing resources on the same cluster. Another aspect is the proportion of ghost cells with respect to the cells without partition, they increase in numbers by increasing the number of partitions, and that reduces the efficiency. In terms of efficiency, one often speaks of scalability, namely, to a given number of processes (N), for a fixed-size problem (DOFS), how much is the speedup (S),

$$S = \frac{1}{p + \frac{1-p}{N}}, \tag{5.8}$$

where p is the imperfection in parallelisation. This is often referred to as Amdahl's law. To measure the efficiency by parallelisation, one measures the total time (T) spent at each cluster. Ideally, these times should be equal, but usually they differ, causing processes to idle waiting for synchronisation, as will be discussed further in detail later. The part of the code/process that benefits from parallelisation, the wall time drops by $\frac{(1-p)T}{S}$, while pT remains unaffected. However, this mostly gives a pessimistic view with increasing N but other conditions and procedures to measure efficiency in parallellisation[109]. To achieve an optimal partition, as a preprocessing step, partition managers is applied. In general, nested gather/scatter seldom impacts efficiency unless there is an outer loop redoing the steps, for example, NP algorithms.

The usage of global vector representation

The global interface of the displacement is the key element that is distributed, which is done in two steps. Firstly, looping over all DOFs and extracting interfaced values and storing them it in a global representation on each processor, which involves MPI gathering. Then locally on each processor, updating solution vector by inserting values of interface boundary, which are later used for next update in the strongly coupled FSI procedure. The extract and insert are executed with the same procedure (i.e. just interchange get_field with set_field) which is advantageous since there is then no need to keep an extra mapping of DOFS index. Of course, within each processor, the DOF index is stored but it is related to gobal vector by using a STL mapping of position index in global vector and the DOF index. Despite the obvious drawback in gathering and scatter, handling the global vector of the interface at every cluster, the timing operation and memory usage determine the necessity for further improvement. The bottleneck at current implementation lies mainly under the hood of the PetSc and solving the linear system and the assemblage. For that reason, no further MPI improvement has deemed being necessary, but with an increasing number of partitions, it is better to avoid global vector representation at each process. The particular reason for handling in this way is because the QR algorithm, used in the residual techniques in the relaxation step, and in the prediction step for the traction, does not scale well with size when using gather/scatter.

5.3 Postprocessing

After the FSI procedure is finished, one obtains sampled data as a function of time. These are probe data in the solid and the fluid, statistics of the convergence, and a measure of proximity which gives an estimate of IE. The statistics of convergence are

in terms of a key indicator of accuracy, namely the maximum of the norm at the final sub-iteration at each timestep, with it's face index on Γ and the largest difference between fluid and cell centre. Apart from the governing fields, the sampled data contains intermediate data such as the residual at Γ, Eqn (2.22), used in the action step relaxation. The analysis of these data is phrased as post-processing and visualised mostly by plotting processed data in terms of non-dimensional units or using Paraview[110] to visualise the governing fields velocity/displacement/pressure or vorticity, also known as the Q plot. The Q plots are coloured by the sign of the U_y velocity or the velocity field (interpolated), which allows rough information on frequency such as angular velocities and direction of change. Paraview also allows ground-truthing data (picking cell values), interpolation in between cells, cutting out cell stripes, and providing plots of governing variables on cells, smoothed surface, or along curves. Each data has an interface, allowing their case description to be imported. Then each solver provides extra data such as drag/lift coefficients, and governing variables at probe points.

5.4 Validation of the FSI procedure

The FSI solver is validated by several steps. It begins by breaking it down into the modules as described by the flow chart Figure 2.2; each step can be tested stand-alone; the essential step is to ensure that to given input, the expected output is generated. Several parts of these steps have to be assumed to be validated by the supplier of the package, but it is still a matter of proper usage and the case of an application that requires validation. Then a comprehensive efficiency study of the scalability is carried out. The validation against experiments and other FSI codes is investigated.

Table 5.4: Validation case of Fluid step for AR 5 cantilever in a channel flow for Re 5000 [ConfI]. Rounding off to two digits due to precision.

Reference	C_d	C_l	St	Remark
ConfI	1.49	0.18	0.11	$St_{KH} \sim 0.5$
ConfII	1.47	0.09	0.11	$U_R = 2$
Exp.[111]	1.42	-	0.11	Re 73000
Numerical[112]	1.61	-	0.11	Re 5000

5.4.1 Fluid step: Validation of OpenFOAM

Several studies have validated OpenFOAM, both experiment and commercial/open source programs such as ANSYS-Fluent[113,114], but one of the founders of OpenFOAM also has a monograph with detailed error analysis[97]. ConfI serves as validation to ensure a proper case setting and usage of the fluid solver. This study involves

a domain study, a study of different discretisation schemes and turbulence models that was performed and applied to principal cantilever configurations in Figure 6.1. TVD scheme on divergence term was the best suited with the application of a blend (0.1). For remaining terms, backward on the temporal and otherwise Gauss linear were shown to be adequate. Further, it was shown that the distance of the cantilever surface shorter than 10D/7D to the inlet/walls produces a signature of pressure, hence implying coupling. ConfI provides the drag/lift coefficient, mean velocity profiles, and frequency spectra, Dynamical Mode Decomposition (DMD). Analysing the turbulence generated by the bluff body. Table 5.4 shows the characteristics of Strouhals and drag/lift coefficients of the bluff-body in channel flow, compared to the experiment of the closest domain setting (same aspect ratio). In general, the frequencies, the shape of the wake, and the slope of the inertia range follow the expected behaviour from experiments, see for example the results of Sumner[115]. The average flow field was identical regarding the size of the wake, presence of vortices, and frequencies, including the intermittency effect as presented in the work by Sumner on clamped-free end[116]. Regarding turbulence modelling, it was observed that the best results were obtained with dynamic one-equation giving a near -5/3 slope in the inertia range at the position with the largest mixing downstream. A mesh dependency study showed that convergence within 10% accuracy of the drag/lift was obtained with 10 cells/D; no further improvement was observed with 30 cells/D. In the enclosed papers, the resolution is 20 cuts/D or better.

5.4.2 Solid step: deal.II

The validation of solid step is normally made against the beam theory, using the theory of elasticity and the analytic formulas. Nonlinear effects are not included, including dependency on the Possion ratio[117]. Solving the balance equation using continuum modelling will thus not provide the same value as the beam theory, but the expected difference is often only parts of a per cent for the amplitude and a few per cent for frequency, the latter mainly due to imprecision in measurement. For the frequency (rad/sec) and max deflection for uniform load upon the width side of a clamped-free/clamped-clamped,

$$f_v = \frac{K_1}{L^2}\sqrt{\frac{EI}{\rho A}}, \delta_m = K_2\frac{pL^4}{EI}, \tag{5.9}$$

where K_1 and K_2 are constants unique for flexuration mode and configuration[118] and EI often phrased as the stiffness but I is the area moment of inertia, the p is the load per unit height, A is the area of cross section. For the first mode of the cantilever $K_1 = 3.52/22.36$ and $K_2 = \frac{1}{8}/\frac{5}{384}$.

The mode shape for the lowest flexuration of the clamped-free,

$$\delta(x) = \frac{px^2}{24EI}(6L^2 - 4Lx + x^2), \tag{5.10}$$

similarly for clamped-clamped,

$$\delta(x) = \frac{px}{24EI}(L^3 - 2Lx^2 + x^3), \tag{5.11}$$

where x is the position in between 0 and L. The damping model can be evaluated by two approaches, either computing the peak resonance height ($R_{d,m}$) from an FHV or by the fraction of decay in a FCS, the amplitude is lowered by the damping ratio (ζ)[119]. The following formulas for computation of the damping ratio,

$$R_{d,m} = \frac{1}{2\zeta\sqrt{1-\zeta^2}}, \frac{1}{n} ln \frac{x(t)}{x(t+\frac{n}{f_v})} = \frac{2\pi\zeta}{\sqrt{1-\zeta^2}}, \tag{5.12}$$

where n is the number of periods between points of measure. To evaluate the damping, one sets the damping ratio, and the Rayleigh model's capability to reproduce that value is measured from simulation. Using the above formulas, the deal.II solver was verified to provide accurate solutions already after 8 cells/D, but the convergence is slow using continuum elements, requiring more than 30 cuts/D to reach within theoretical accuracy with respect to nonlinear effects.

Numerous other time marching techniques exist[120]: the central difference, explicit as well as implicit, conservative, and dissipative. A variant of GN22 is HHT-α[38]. Another commonly applied method is the α method[37]. In addition an implicit scheme with an extra half step[121]. From an evaluation, it was observed that for the partitioned technique, as long one use a second-order accurate scheme, the choice of the scheme was irrelevant. GN22 procedure is by far the most cited in the literature for partitioned FSI and therefore selected as a time marching procedure. In addition, RD modelling suffers from being unstable for displacement driven application, that is, instead of applying traction, one apply Dirichlet condition, in solving the equation of motion, the solution to this is to implement a visco-elastic model (VE), which is not covered in this thesis.

5.4.3 Mesh motion step

This should ensure that the ascribed change is obtained for a given displacement field; it was a functional test, using ascribed displacement and direct downloading data and studying the result in ParaView and ground-truth data by debug trap, see Figure 5.7, for the thin-beam experiment, presented in paper I. In this figure, the surface from the

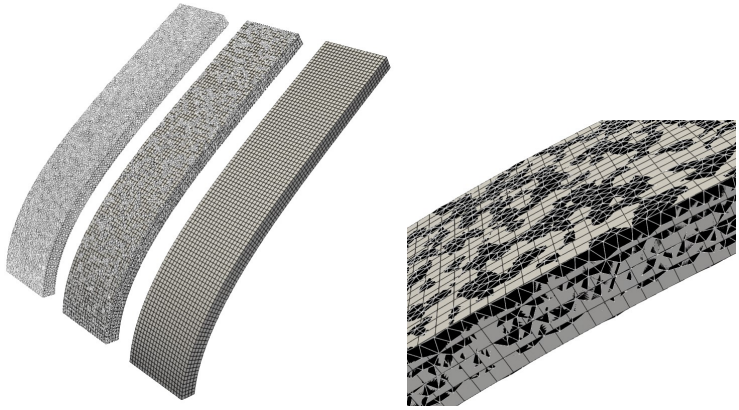

Figure 5.7: The surface of the thin-beam [D, d, L]=[0.01,0.002,0.05]. The observed fluctuations are of the order 10^{-5}, this behaviour is an effect of non-matched fluid centre onto a solid surface. It's a general observation that this pattern is a function of cell size, thus increasing, will reduce the deviation further.

fluid surface is placed the furthest away with black surface colour and white segments of facet segments, and otherway around for the solid, closest to view, the middle is superimposed, showing the difference. Note, although the facets cell centres are to the order of 10^{-8} in l^2 norm, it was sensitive and could crash while further increasing tolerances, the displacementLaplacian mesh motion solver. Since it was observed user-defined error by patch assign, proximity analysis is performed at the start of each simulation, which measures the largest deviation between source points ascribed by patch fluid and solid interface. This ensures that the transfer between the solid and mesh motion steps is consistent in each simulation. Also, a flag is introduced as a warning, issued whenever missing matching occurs, i.e. a cell lacks a correspondence at the partition level; this applies to the mapping in both directions. A simulation is discarded if any unit test fails and/or issued a flag during run-time or proximity of matching fails.

5.4.4 MPI scalability

The LUNARC HPC Aurora consists of 180 nodes, each node 20 threads (40 processes). Due to interaction with other users between/within nodes, the scalability test was limited to up to 60 threads, and the nodes were confined to one rack since otherwise was biased by competition over bandwidth but also by limitation in Mesh motion solver at that target version; since using reduce all call lowers the efficiency for the mesh motion significantly already at 80 threads and becomes negative at 120 threads (fluid step stand-alone). See Figure 5.8. Scalability is also lessened due to unstructured mesh and using parMetis, giving a disadvantage compared to structured mesh, increas-

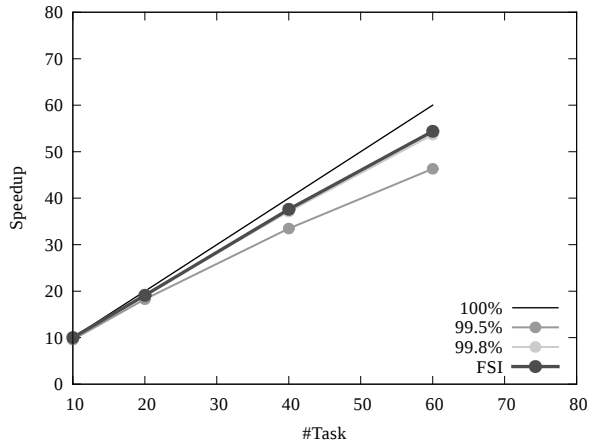

Figure 5.8: The speedup with estimated non-parallelisable $p \sim 0.2\%$, the Y axis is the scaling and the X axis the number of processes.

ing with decreasing cells/threads, below a certain point, the overhead and variance of the number of cells per thread hampers the analysis and increasing the total number of cells only would make the lower range costly and hence a negative scaling is observed, see Figure This scalability study provides an estimate of the non-parallelisable fraction (p) by using the Amdahls law, a fit to Eqn (5.8), gave $p \sim 0.2\%$. The application is the quiescence flow study from paper III, for two nodes, the speedup is around 37 of N=40. However, that is for a run with no competition over bandwidth; in practice, for an average load of users, the efficiency for two nodes may drop down to a scaling of around 30. For current application with cantilever with AR and width to thickness

Table 5.5: modified Richter case

Reference	X/D	f_1/Hz	f_2/Hz	f_3/Hz
This work	0.175	2 ± 0.5	7.8 ± 0.5	26 ± 1
Numerical [3]	0.18	2	7.3	27

ratio, for a typical mesh of 3 M cells for fluid, and around 50 000 cells for the solid, leaving to around few thousands of facets on the coupled surface, the majority of time is spent at fluid step (94%) and solid step comes in second place (4%), then there is 1.5% outside timing statistics, while time for the relaxation step is negligible (0.5%). The expected outcome for 40 processes (2 Intel Xeon E5-2650) in one week (168 h) is around 17000 subiterations, thus providing, assuming mean 4 subiterations and a time step of 2 ms, a total simulation time of around 8.5 seconds. This normally for an FSI application using rubber/water material with around 1-2 Hz ground frequency, generates around 10 shedding vorticies.

5.4.5 FSI validation

There are standard benchmark cases, although not relevant to the current scope of this study; these have been reviewed and studied but are not presented here. For the current case setting, two benchmarks are applied. Firstly, the experiments of Luhar et al[1], see the paper I, Table 1 and Figure 8, provide validation against experiments and numerical study. Secondly, in the literature referred to as the Richter case, an elastic block immersed in a channel, with AR 2:1 for height to thickness and 4:1 with respect to width to thickness. The thickness is set to D=0.2 m. Using the case setting from the numerical study that is benchmarked against[3], see Table 5.5 for the tip displacement and the three dominant frequencies in displacement. Figure 5.9 depicts the velocity profile from midplane at three seconds simulation time. Since the simulation will, in the end, fail by strong resonance between inlet and outlet with respect to the solid.

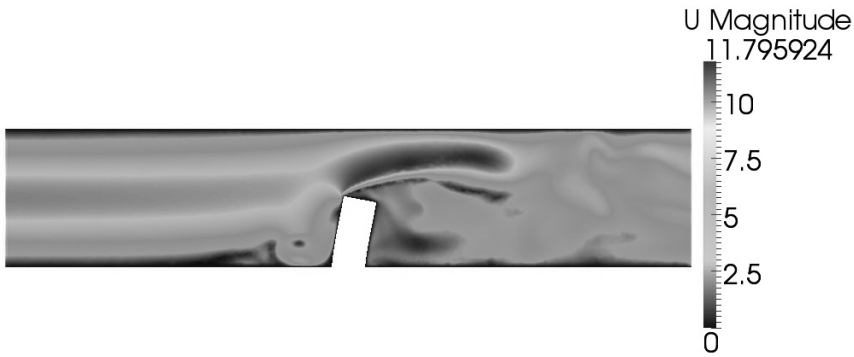

Figure 5.9: modified Richter case, see Ref[3] for the case description. Midcut magnitude velocity at 2 seconds simulation.

Chapter 6

Summary of results

The enclosed papers contain detailed results; It begins with a case description, with the principal cantilever configurations and the mesh applied. The validation results are presented and placed in context with other sources. Thereafter stability analysis and efficiency analysis of the FSI solver are presented. The chapter ends with the presentation of a case which is part of the outlook of this thesis.

6.1 Case description

The thesis is focused on channel and still tank (quiescence flow), with an immersed cantilever that is either clamped-clamped or clamped-free; see Figure 6.1. The fluid mesh is unstructured since it has been shown in technical reports, available on CFD forums on the subject strategy to move mesh with OpenFOAM using a structured mesh with the mesh motion solver with best-displayed scalability MPI wise often requires re-meshing during the simulation. That functionality was not feasible with

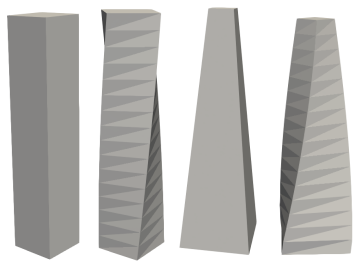

Figure 6.1: Principal configuration of the cantilever with different shapes, from left to right: straight, twisted, tapered and twisted-tapered. [Conf I]

the current application version for various reasons. Using non-dimensional numbers enables flexibility and provides a compact description for a flow case. For all cases, the width is always 1D; the cantilever is therefore defined by the aspect ratio height to thickness (AR). The applied spatial discretisation on fluid is TVD for convective, backward on temporal and gauss linear on the other terms in governing equations. No wall modelling is applied, a nearly DNS, with $y^+ < 1$, where y^+ is a dimensionless number telling the resolution of the boundary layer at the wall. At the interface 20 cuts/D unless otherwise said. Then the mesh is stretched (at most 20% at adjacent

cell layer, on average 5%) to ensure sufficient division of the space for the forming and moving of the vortices generated by the cantilever.

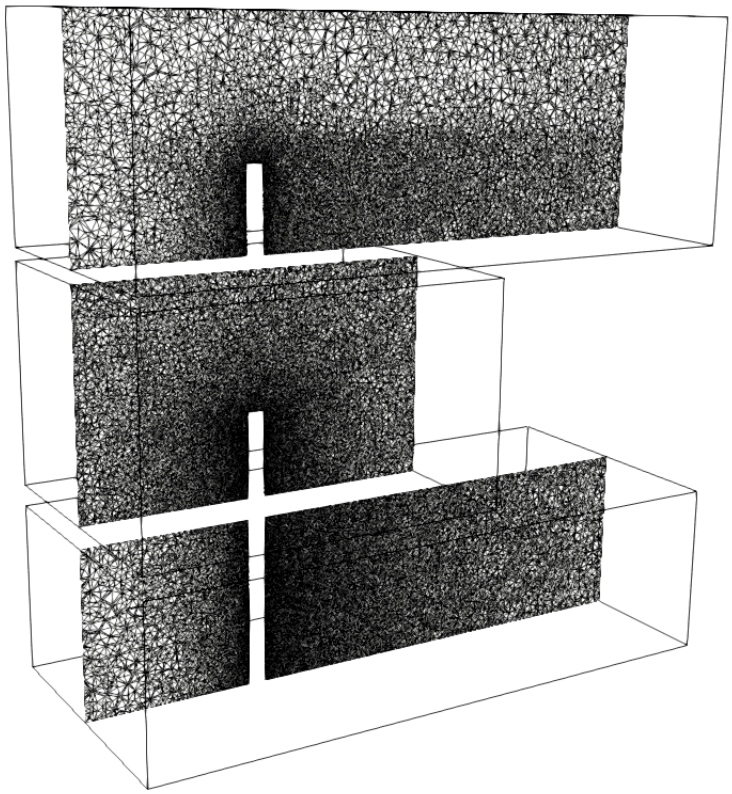

Figure 6.2: Principal mesh description. The upper is for channel flow, with refinement in the wake and around the cantilever. The middle is for quiescence flow case, with only refinement around the principal cantilever, the lower is clamped-clamped cantilever case in channel flow.

Therefore the channel has an additional mesh strip to cover the downstream vortex street, and still tank has not. Throughout the chapter, distances are measured by the ratio of the thickness of the cantilever (D). The channel flow is velocity driven, where the BC of the opposing opening uses prescribed velocity and zero gradient pressure at the inlet and at the outlet, a pressure gauge with zero velocity gradient. The setting of aspect ratio, material and conditions, and Reynolds number set the parameters needed to reproduce the data. Unless otherwise specified, as by reference or explicit in text/figure, the mass ratio is 1, and the cantilevers Young modulus is E=59 MPa. For quiescence flow, traction is applied to the side that faces the opening, and the openings are set with gauge pressure. All walls have a non-slip condition.

6.2 Validation

Paper I and Paper III contain the main parts of validating the FSI solver. During the work, before the publication, conference papers ConfI and ConfII include the preliminary results for the coming papers I and III. A particular validation in this context is the Solid-Solid Interaction[21], phrased as "interesting but not particularly effective", but as a validation, most useful, since it is a black-box and case-specific. The strategy is to replace the fluid solver with a solid solver for the case of interest, split the solid domain, and apply traction over a surface part that is split, and the reference is merely the "monolithic", i.e. no split, see Figure 3 for case setup and Figure 4 for the absolute error versus number of cuts per D presented in paper I, using the formulas presented in section 5.4.2. This validates the traction transfer for a static deflection case, showing the difference between monolithic and partitioned in absolute error, decreasing to 0 with increasing resolution. It mainly sets the resolution and IE error from the log file in perspective as an accuracy index, see Figure 5 in paper I. As long

Table 6.1: Luhar experiment. Re 1600

Reference	C_d	X	Z	Remark
Paper I	1.12	2.30	-0.65	constant Smagorisnky
Paper I	1.11	2.18	-0.67	dynamic one-equation
Exp[1]	1.15	2.14	-0.59	Error $\sim$10%
Numerical[77]	1.05	2.45	-0.75	constant Smagorinsky

as one has 0.2% or less IE, the partition provides a solution error of a few percent or less, requiring an average of 20 cuts per D at the surface. The deflection/frequency and RD modelling for the solid-state solver are also compared to beam theory, see Table 6.2, showing static/dynamic deflection. The RD modelling can only be ratified by setting the damping ratio and measuring its outcome. The OpenFOAM state solver

Table 6.2: validation of the solid step by applying traction to a cantilever side. AR 5

Reference	A_x/m	f/Hz	ζ
Paper III	$0.00799\ m$	$7.95 \pm 0.15\ Hz$	0.049
Beam theory	$0.00796\ m$	$7.83\ Hz$	0.05

is discussed in the previous chapter. For an additional validation of the dynamics, apart from the Richter case Figure 5.9, see Table 6.1, which shows the deflection in the experiment by Luhar et al[1], as depicted in right Figure 1.2 should be compared to Figure 7 in the paper I. Another validation aspect is reproducing the flow topologies observed in the literature, such as the VIV, the shedding frequency, the deflection limits and distances between pin vortices. An eluded pattern is the upper branch; only the lower branch was observed in paper I with a deflection limit of mostly around 0.8 and Strouhal's number around 0.1, see Figure 8.

6.3 Stability analysis

Paper II is mainly focused on FSI stability for the immersed cantilever. Firstly a classification of how the FSI solver fails is introduced, since for unstable FSI using PICARD, relaxation increases the stability but may produce a defect solution, and as observed, it is manifested differently. An FSI solution is stable when it produces a bounded solution throughout the simulation. Then the influence of the action steps to stability and its parameter dependency are presented.

Definition of failures

Three types of failures can occur in the context of a coupling strategy in numerical simulation. Type I failure occurs when the coupling leads to an increasing norm, despite each subcycle of the FSI having a stable result, thus leading either to divergence within a few macro iterations or slowly diverging, all depending on the settings in the application. Type II failure occurs when the coupling is stable, and depending on the choice of tolerance, and the frequency of this failure, an error in the solution is introduced by various degrees. Type III failure occurs when the mesh motion solver fails, which mostly leads to type I failure. This study eliminates type III failure by ensuring a proper case description. This classification of failures helps to understand and improve the methodology's stability.

The influence of action step to stability

From paper II, using traction SUR as a reference, i.e. continuation over the traction from the previous to the current time step, continuation over Young modulus and mass show a relative difference of $-0.26 \pm 0.5\%$, IQN-ILS[5] $0.17 \pm 0.3\%$, while FIR applied to the traction, $-1.3 \pm 1.8\%$. The application is an immersed cantilever in channel flow with R=2, with the PICARD unstable at Re 500. Hence action step continuation and FIR stabilise the FSI, and continuation provides similar accuracy as compared to relaxation, showing that the solution is not altered by either of the methods. Therefore, these action steps can be applied to stabilise the coupling and improve the convergence of the solution in a numerical simulation using the PICARD method. However, the FIR action step has the drawback of introducing a growing phase error proportional to the width of the filter. Similarly, as the continuation technique can also remove higher-order modes that should be part of the final solution, hence acting like a projection. The same goes for IQN-ILS and R-family[76], if the last iteration is without relaxation[56]. The IQN-ILS-R technique increases stability by

suppressing the zig-zag pattern described by Figure 5 in paper II, but not to the same extent as continuation. Since at the beginning of the strongly coupled FSI iteration, the R-family re-uses previous macro iterations residuals, every iteration is with relaxation and full-matrix. But the continuation technique is more efficient in suppressing this pattern and can be combined with IQN-ILS, providing the most stable procedure in this study.

R versus AR dependency

In paper II, one studies the critical mass ratio R^* at which type I failures occur, the change of R is done by keeping fluid density constant and lowering the solid density. Several studies, see references in the introduction of paper II, have shown that this limit, in terms of vanishing time step, is independent of the structural stiffness, hence the aspect ratio, and the viscosity. It is merely dependent on the added mass effect. For cantilever, inviscid theory, i.e. potential theory, it can be shown that the inviscid added mass is equal to $R^*\Gamma$. Hence, by computing $R^*\Gamma$, it is shown in paper II that the stability is independent of aspect ratio height over thickness (AR) for SUR and PICARD. However, there is no similar theoretical study, showing the stability for IQN-ILS or residual technique in general, except for IGMRES. Paper II, however, implies that despite the improvement in stability by the residual relaxation techniques, all studied residual relaxation techniques fail with the increasing mass ratio for a given aspect ratio. However, the hybridisation technique, a combination of continuation and IQN-ILS appears to give unconditional stability, until between 10 to 20 AR, it begins to fail, hence providing a R^*, it alters the character of failure, going from type I to type II. What causes this was not clear, but a similar behaviour was observed when altering fluid density instead. The stability limit shifts from type I to type II, and often to slightly less R^* value, but less than 1 in difference.

6.4 Relaxation techniques

This section focuses on evaluating the performance of the relaxation techniques used in solving the fixed-point iterative problem for fluid-structure interaction (FSI). Paper IV provides a more detailed analysis of each method. The first aspect discussed is the process of quality assurance, which includes the analysis of probe data and the criteria for selecting the result of relaxation. This is followed by a comprehensive performance study that presents statistics on the best-performing relaxation techniques. The section concludes by providing more in-depth information on the top-performing relaxation technique in the study.

Performance

In solving the fluid-structure interaction (FSI) problem, relaxation techniques can be used to stabilise the solution. The performance of these techniques is assessed by looking at the average number of subiterations required per macroiteration, the standard deviation with respect to the mean value and the fraction of type II failure. The two-step method applies the Gauss-Seidel method at both the solid and fluid response, with relaxation applied at each step. See the results in Figure 6.3. Evaluation results show hybrid/bombardment techniques (HYBRID), using X=2, perform better than standard techniques while blending techniques are the second best. Blending techniques are better suited for alternating series (zig-zag patter, see paper II) unless applied on every even/odd subset; see further in [122] and reference therein. Acceleration techniques such as VEA and TEA have failed to perform better than AITKEN and have a similar performance to residual techniques like IQN-ILS. Methods marked with the super-index "*" generally lead to improved performance, indicating that excluding the first response from relaxation results in better convergence. This finding aligns with previous research that showed that the family of Anderson relaxation techniques performs better for sequences with linear convergence than for those with quadratic convergence [60]. The convergence rate between the first two iterations is nearly two. The DSUR technique is the best-performing non-hybrid relaxation technique, but further studies on other cases are needed to confirm this observation. DSUR should be regarded as a continuance of the work by Degroote with the so-called IQN-ILS-R-family methods. Instead of reusing previous residuals, one instead computes the statistics of relaxation parameters for current subiteration level and reuses them instead applied to the current residual.

A previous study found that AITKEN performed worse than IQN-ILS, but this conclusion might change if the pressure response was studied, as the authors concluded that AITKEN was unstable for those settings. Improvement of AITKEN by replacing the scalar product over the interface with scalar product over the entire field significantly improves the performance.

That a method can be categorised differently depending on the number of used iterations obscures the analysis and opens the question of whether to take advantage of the fact that some techniques work better at different stages of the coupling iteration. This led to the subsequent study, the bombardment technique, a methodology that emerged from this performance study and outperformed all other in this study.

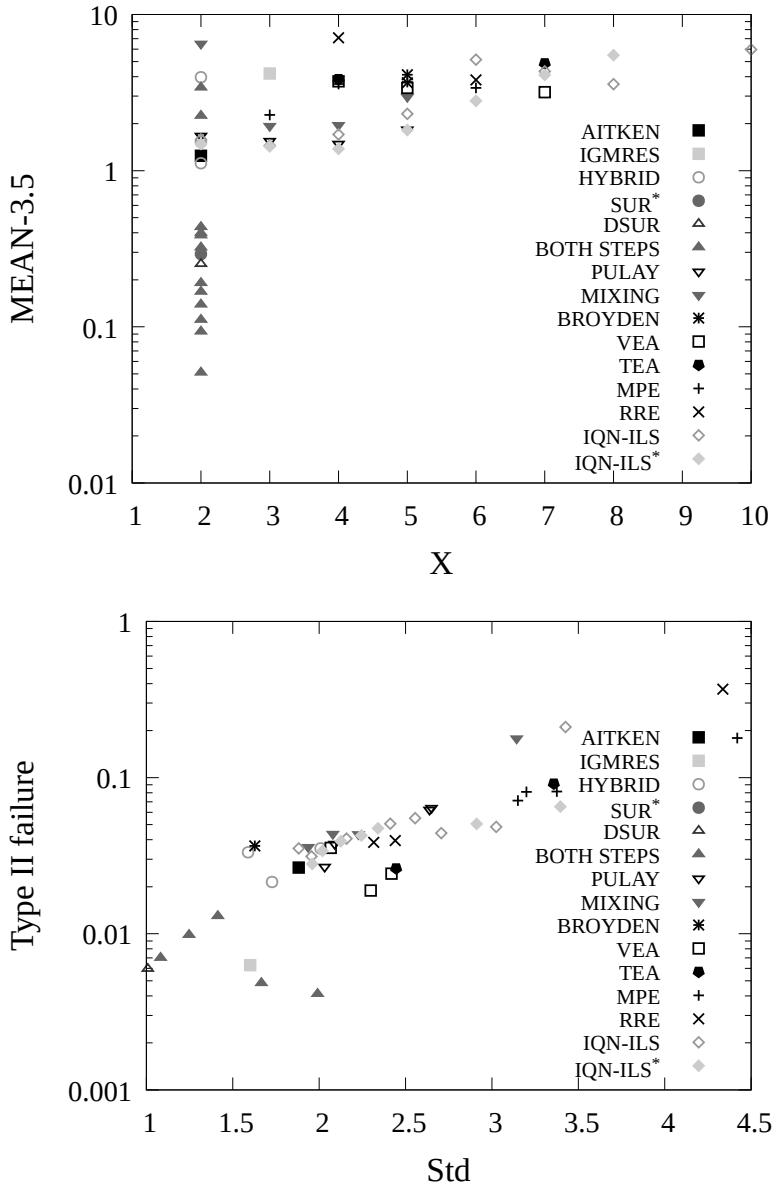

Figure 6.3: Performance statistics with log y scale. Upper: The mean value (with 3.5 subtracted) as a function of the number of X. Lower: Std versus Type II failure.

Bombardment technique

The bombardment technique developed has three evaluation options: selecting the method with the smallest norm, taking the average of all relaxation techniques, or

using the relaxation result as input for the PULAY method. The results showed that selecting the method with the smallest norm was the best choice. Another aspect of interest is to see how the techniques perform as the mass ratio, or added mass, increases. See Figure 6.4 including a comparison to regular relaxation methods statistics.

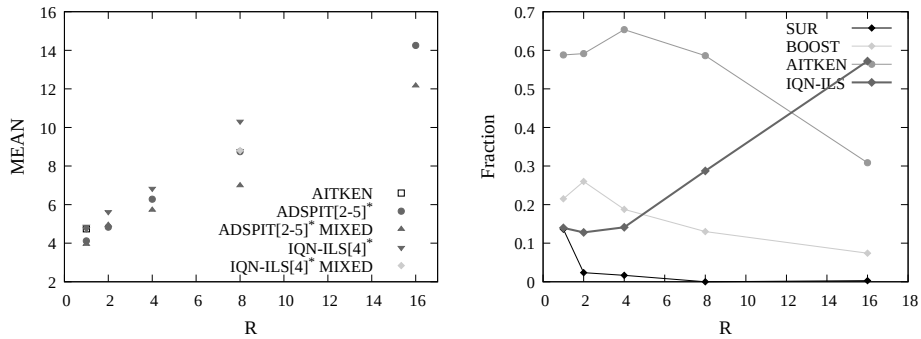

Figure 6.4: BOMBARDMENT Left: Mean value as a function of mass ratio R, with reference to the most efficient stand-alone. Right: selected methods as a function of mass ratio. Showing how residual techniques gain influence with increasing mass ratio. The R=16, IQN-ILS stand-alone are not stable for X=4,5

From paper IV, it is further shown that, which aligns well with the previous section of the performance, the blending techniques based upon two iterations decrease with increasing mass ratio, and the optimised blending decreases with the increasing variance of the number of iterations. MIXING is constantly lesser performance than IQN-ILS, by only a margin though, (within a factor of 10 in norm) but it has been the other way around for case settings with a clamped-clamped cantilever, especially while the continuum wave is amplified. Using double end-condition ($l^{\infty} < 10\epsilon, \frac{l^2}{\sqrt{n}} < \epsilon$) as described in paper IV is called "MIXED".

6.5 Application: clamped-clamped cantilever

The clamped-clamped cantilever immersed in a velocity-driven channel is studied. Although the primary goal of this study is to investigate the statistics of the BOMBARDMENT technique, a secondary goal emerged, to investigate the upper branch, which has been eluding earlier studies and thus not part of the enclosed papers. The triggering of the upper and lower branches is obtained experimentally by slowly changing the inlet from upper/lower speed. The most strongly dependent parameter in the fluid for efficiency and stability is the mass ratio. To evaluate the BOMBARDMENT, a test series is created with relaxation ADSPIT[2-5]*, where two simulations are executed for each mass ratio: changing inlet velocity from lower to higher, and vice-versa.

Case setup

At first, the application applied uniform constant inlet, three different heights such that the same structural damping is achieved with mass ratio 1. See upper Figure 6.5 for an instantaneous Q plot with all cantilevers aligned and isosurface coloured by the sign of U_y showing the rotation of the vortex tubes. The immersed frequency is

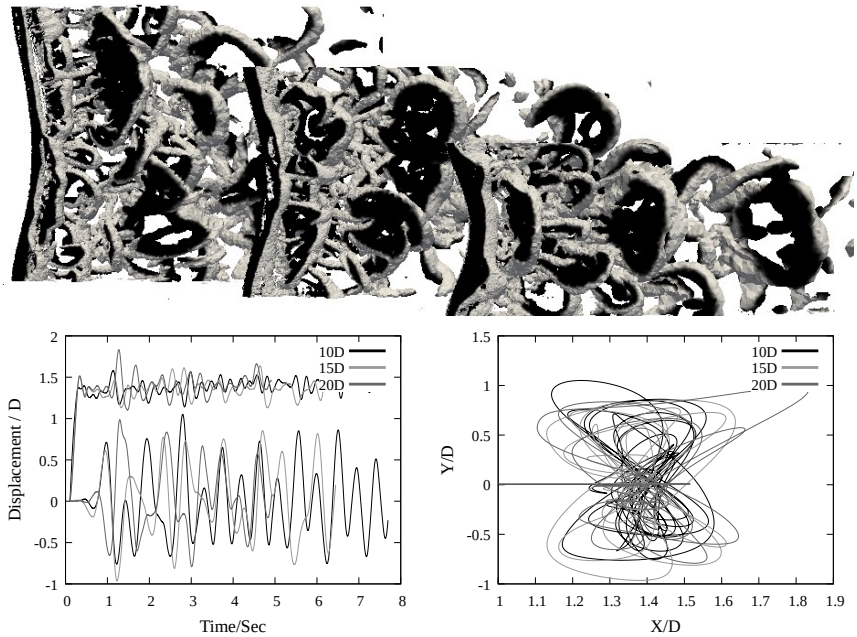

Figure 6.5: Upper: Clamped-clamped cantilever in a velocity-driven channel, uniform flow with inlet U=4 m/s. Lower Left: Inline (X) and cross-flow (Y) displacement scaled with D. Lower Right: phase plot of the left data

around 2 Hz. The settings differ in such a way that clamped-clamped can have standing continuum waves. To alleviate the eventual build-up of these waves, RD damping is applied, specifically, β damping, a common way to suppress high-frequency modes, corresponding to a damping ratio of 0.025 at 2 Hz. The constant inlet at synchronisation point provided insight in the settings for the test serie, see lower part of figure 6.5. The simulation begins with a 2 second interval initialisation with the starting inlet, then linearly for 10 seconds changes from inlet from 2 m/s to 6 m/s. The FSI begins after 0.1 seconds. The corresponding simulation from higher to lower differs in that the initialisation is lowered by 1 second. In the first test series, the inline deflection was allowed, which at first appeared to trigger a larger envelope but due to the collapsing of vortex tubes, failed to maintain the envelope. The remedy for this appears to be freezing the degrees of freedom in inline deflection. As a result, a hysteresis

appears, showing the difference in the envelope is large between the simulation going from low to high, versus high to low, inlet. Although the frequency changes linearly with inlet velocity at inlet velocity in presynchronisation range, at the higher inlet, i.e. at synchronisation/desynchronisation, the cross-flow frequency appears to be locked and the amplitude reaches to a plateau. This may implicate that the upper branch is triggered. The simulation with frozen displacement inline revealed an issue with triggering the secondary mode of vibration (∼4 Hz). This required a change from uniform inlet to elliptic inlet profile. Still, one can see partially this mode being activated but to a lesser extent with increasing height, which is why the study presented in left Figure 6.6 is restricted to height 15L, the 20L case is more costly although showing the same result and 10L has a larger proportion of second mode than the other cases. Further, which is more interesting, the maximum amplitude (scaled with D) increases with increasing mass ratio. At R=1, it is around 1.15; at R=8, it is around 2.3.

The BOMBARDMENT technique revisited for clamped-clamped cantilever

For that test series, using elliptic inlet, height 15 L, by varying mass ratio from 1 to 8, using frozen inline, the statistics of subiterations and selection of relaxation were sampled and their results are presented in the Figure 6.6 - 6.7. The primary goal is to

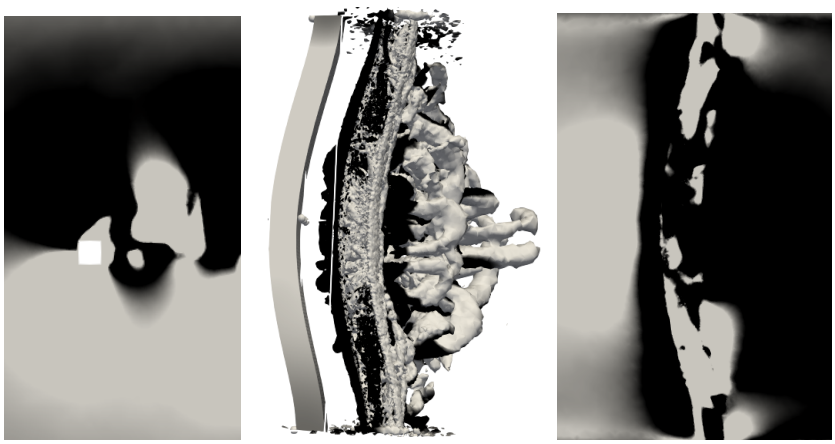

Figure 6.6: The clamped-clamped cantilever with R=8, sweeping inlet velocity, elliptic inlet profile. U_y signed coloured with thresold of 0.1. The motion of the cantilever is decreasing amplitude in the Y direction. Left: XZ plane at height Z=7.5D front side of the cantilever. Showing coherent flow with negative direction (down). Middle Q plot: the cantilever, moved 0.5 m sideway in the Y direction, showing the vorticity and the motion in the Y direction, note the same sign where it is strongest. Right: From the cantilever behind, showing at the maximum deflection, turning point, the black part is the motion of the fluid moving in Y direction to the right.

study the BOMBARDMENT statistics; from R=1 to 8, the mean value of subiterations goes from 3.6 to 7.1, fully consistent with the results in paper IV, but the selected

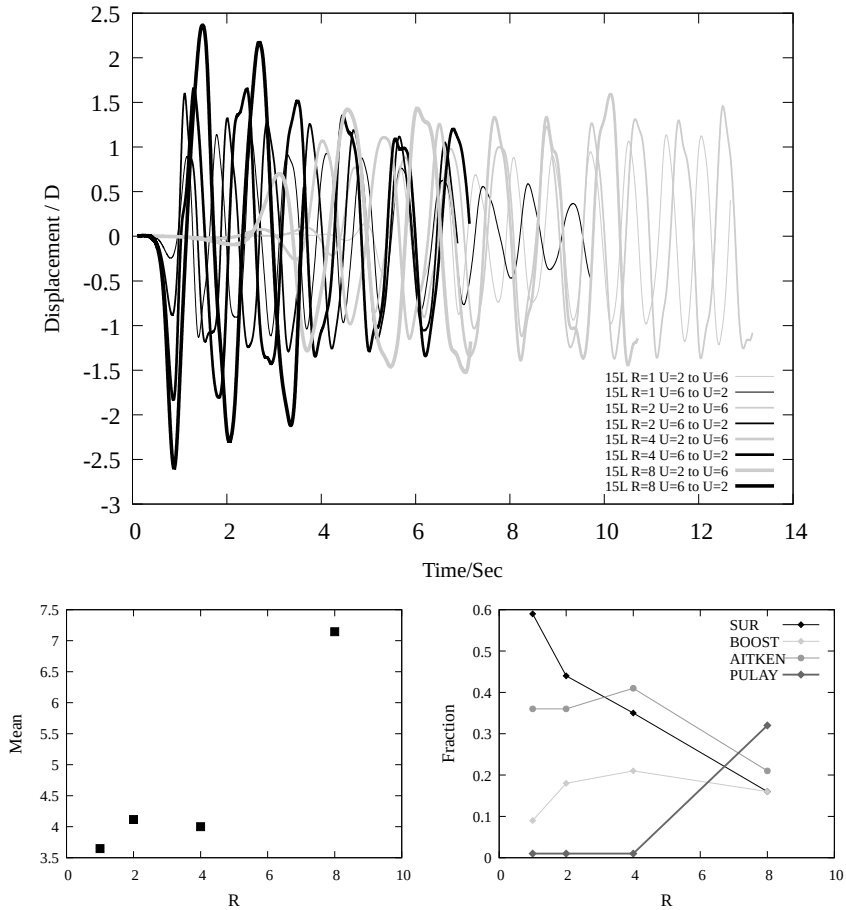

Figure 6.7: The clamped-clamped cantilever with R=8, sweeping inlet velocity, elliptic inlet profile, R=1 to R=8, increasing linearly from 2 m/s to 6 m/s. Upper: The hysteresis, cross-flow deflection. Lower Left: Convergence statistics. Left: average iterations. Lower Right: Selection statistics

statistics differ significantly; see lower right Figure 6.7, the PULAY is the leading selection for R=8, all methods less than 0.01 fraction selected are omitted. From the Figure 6.6, the Q plot shows that the vortex tubes, throughout the simulation alter in size, but never collapse. This implies that it is indeed a significant aspect in triggering the upper branch. It also explains why this phenomenon is only observed for long slender cylinders designed so it minimises the inline deflection [45]. Numerous more results on the aspects of flow topology are observed, one such of interest is the presence of mode B. A study of the transition in the modes shows that mode A appears around Re 180, then around Re 400, it changed into mode B. The indicator for the mode is the distance between the release of pin vortices, span-wise.

Chapter 7

Conclusions and outlook

The study focuses on conducting a thorough investigation of the dependence of parameters on fluid-structure interaction (FSI) in partitioned systems, with specific emphasis on cantilever beams in both channel and quiescence flow. The objective is to understand the impact of fluid flow on the structural behavior of cantilever beams and how different parameters affect the overall system dynamics. The study aims to provide insights into the design and optimisation of cantilever-based systems for practical applications, particularly hydroelastic. All arguments here is not fully covered in the compilation thesis, some can be found in enclosed paper, therefore mentioning in parenthesis.

7.1 Conclusion

The FSI methodology outlined in this thesis has shown to be robust, and efficient, to the same level or better as presented in the literature. This is based upon two arguments, firstly by reproducing benchmarks from references, and secondly, by comparing the methods on the current application, cantilever immersed incompressible flow, together with parameter study of the stability limits and with that the efficiency in convergence statistics. It has been shown by reference and in this thesis that the pressure field for incompressible flow is a significant field and cannot be omitted in a study (paper III/paper IV). It is rather concerning that this consistent pattern of omission of pressure is the dominant feature in articles, it's the big elephant in the room. For that reason, separate supplementary data has been provided in paper IV. But with the current methodology at hand, and implementing similar partitioned FSI as in comparative methodology, the split traction approach with resetting all fields except interface at every subiteration appears to be crucial in achieving increased efficiency, not stability. Otherwise, the artificial growth of residual limits the partitioned FSI methodology to moderate mass ratio/structural form factor (e.g. aspect ratio width to height, AR) (paper II). The most stabilising technique so far is the continuation technique, but less efficient. The continuum waves have proven to be a challenge due to the requirement of high resolution and impairment to efficiency, the most commonly observed resolution to this in the literature, is to either by material eliminate its source or by implict filtering it out by choice of time step. (paper I,paper II/paper IV). In bullet form, summarising conclusions,

- The partition FSI methodology for incompressible flow is limited by the product of the structural factor and mass ratio. But relaxation techniques so far do not alleviate the stability issue and thus with increasing limiting factor, i.e. mass ratio or structural factor, the partitioned technique will ultimately fail. The second governing parameter appears to be the stiffness, which closely relates to the accuracy of the transferred data, phasing/interpolation error, since with decreasing stiffness with the same traction, a smaller time step is required, hence requiring higher accuracy. (paper II).

- The continuation technique is the most stable technique and can either be used as a precondition or relaxation technique itself. (paper I/paperII)

- The blending technique phrased as the DSUR has shown promising results and is for stable PICARD the most efficient technique. (paper IV)

- Improvement of AITKEN by replacing the scalar product over the interface is more efficient than IQN-ILS and has a stability range similar to IQN-ILS-4 (paper II).

- The residual technique is only first order in rate of convergence with respect to the norm over a number of subiterations, it is outperformed by the accelerating technique of modified AITKEN and the DSUR, notably for those case settings for which PICARD is stable, but as PICARDS grows unstable, the residual technique becomes the better choice, notable IQN-ILS with skipping first three iterations. (paper IV)

- By computing the residuals simultaneously and selecting the technique with lowest norm and apply its solution to interface, the BOMBARDMENT technique. Further, by having two sets of relaxation techniques, applying the first up to given subiteration, then the second after, this is the split technique. Together these two techniques, provided the most efficient and stable technique with only tolerance as input. (paper IV)

- The partitioned technique is highly scalable for parallelisation and appears to be limited by each solver stand-alone, this allows the partitioned technique to be applied to HPPC and thus study larger applications.

Although the BOMBARDMENT is a black-box technique, the choice of FSI tolerance for the end criteria and type of norm sets the limit of the technique. The l^2 norm is the most efficient, but this study shows that the l^∞ is the most accurate, by identifying the critical tolerance for accuracy for each norm, the combination of the norms, requiring both, provides the best marriage between accuracy and efficiency (paper IV).

In addition to the conclusions in methodology findings, some interesting flow topology and parameter dependency were to be found. First and most importantly, the turbulence, the chaotic motion with its characteristic degrade with a cascade of motion was successfully modeled by LES providing a turbulence spectrum that shows promising results (paper I) but is still inconclusive whether consistency is achieved near the resolution limit, fortunately, the noise generated by FSI can be reduced by further increasing resolution in both time and space, but the cost in computing resource limits its value. A functional relation between drag force and the form factor provides a theoretical expression for the change in drag as a function of the reduced height (paper I). Another observation is that semi-empirical methods for some MEMS devices would benefit from reparametrisation, due to the presence of convective flow (paper III). Further, one aspect of vortex-induced vibration is the cause for limiting of the cross-flow deflection, it appears to be related to the collapsing of vortex tubes attached to the cantilever, providing only lower branch DLF graphs with a value around 0.8 reduced deflection for R=1, while freezing the inline deflection, it quickly increases with increasing mass ratio and shows hysteresis and frequency lock, however, no apparent frequency shift for current case setting.

7.2 Outlook

The study had one significant deficiency: the mesh motion technique. The characterisation of the failures omitted the third type. The limitation of a diffusive motion solver limits the application. Therefore, extending the functionality with other motion solvers would be an immediate improvement. Another limitation is the ALE/unstructured mesh restriction since using the same partitioned approach to structured mesh and IB procedure would increase the application range. The main reason to apply a structured mesh is to increase the accuracy. This would also allow further study the impact of turbulence to FSI. Using IB would allow extension to other applications. Another point of interest is to explore the impact of the findings from relaxations to the relaxation methods applied in computational chemistry. During the implementation, it was also realised that an interface code would benefit much if a standard exchange protocol could be introduced, allowing schema to create the necessary digital handshakes between stand-alone solvers. Also, providing the means to generate coupling schema based upon scripting, increases the application range in using partitioned techniques. This since different application may required different coupling strategies. Some commercial solvers already offer this, but with a standard exchange protocol, this would significantly facilitate the integration of stand-alone solvers.

References

[1] M. Luhar and H. M. Nepf, "Flow-induced reconfiguration of buoyant and flexible aquatic vegetation," *Limnology and Oceanography*, vol. 56, no. 6, pp. 2003–2017, 2011.

[2] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodríguez-Ferran, *Arbitrary Lagrangian–Eulerian Methods*, pp. 1–23. John Wiley & Sons, Ltd, 2017.

[3] K. Rege and B. H. Hjertager, "Application of foam-extend on turbulent fluid-structure interaction," *IOP Conference Series: Materials Science and Engineering*, vol. 276, no. 1, p. 012031, 2017.

[4] C. S. Peskin, "Flow patterns around heart valves: A numerical method," *Journal of Computational Physics*, vol. 10, no. 2, pp. 252–271, 1972.

[5] T. E. Tezduyar, S. Sathe, R. Keedy, and K. Stein, "Space-time finite element techniques for computation of fluid-structure interactions," *Comput. Methods Appl. Mech. Engrg.*, vol. 195, pp. 2002–2027, 2006.

[6] C. Antoci, M. Gallati, and S. Sibilla, "Numerical simulation of fluid-structure interaction by SPH," *Computers and Structures*, vol. 85, pp. 879–890, 2007.

[7] A. Bhadauria, B. Dorschner, and I. Karlin, "Lattice Boltzmann method for fluid–structure interaction in compressible flow," *Physics of Fluids*, vol. 33, no. 10, p. 106111, 2021.

[8] Z. Ozdemir, M. Souli, and Y. M. Fahjan, "FSI methods for seismic analysis of sloshing tank problems," *Mécanique & Industries*, vol. 11, no. 2, pp. 133–147, 2010.

[9] C. Conca, A. Osses, and J. Planchard, "Added mass and damping in fluid-structure interaction," *Computer Methods in Applied Mechanics and Engineering*, vol. 146, no. 3–4, pp. 387 – 405, 1997.

[10] A. Tijsseling, "Fluid-structure interaction in liquid-filled pipe systems: A review," *Journal of Fluids and Structures*, vol. 10, no. 2, pp. 109–146, 1996.

[11] J. Morison, J. Johnson, and S. Schaaf, "The Force Exerted by Surface Waves on Piles," *Journal of Petroleum Technology*, vol. 2, no. 05, pp. 149–154, 1950.

[12] S. B. Pope, *Turbulent Flows*. Cambridge University Press, 2000.

[13] O. Vasilyev, D. Goldstein, G. De Stefano, D. Bodony, D. You, and L. Shunn, "Assessment of local dynamic subgrid-scale models for stochastic coherent

adaptive large eddy simulation," *Proceedings of the 2006 Summer Program*, pp. 139–150, 2006.

[14] A. Aspden, N. Nikiforakis, S. Dalziel, and J. B. Bell, "Analysis of implicit LES methods," *Comm. App. Math and Comp. Sci.*, vol. 3, no. 1, pp. 103–125, 2008.

[15] F. G. Schmitt, "About boussinesq's turbulent viscosity hypothesis: historical remarks and direct evaluation of its validity," *Comptes Rendus Mecanique*, vol. 335, no. 9-10, pp. 617–627, 2007.

[16] V. Moureau, O. Vasilyev, C. Angelberger, and T. Poinsot, "Commuation errors in large eddy simulation on moving grids : Application to piston engine flows," *Proceedings of the Summer Program 2004, Center for Turbulence Research, NASA AMES/Stanford University USA*, pp. 157–168.

[17] R. M. Darlington, T. L. McAbee, and G. Rodrigue, "Large eddy simulation and ALE mesh motion in Rayleigh-Taylor instability," *Computer Physics Communications*, vol. 144, pp. 261–276, 2002.

[18] C. Wood, A. Gil, O. Hassan, and J. Bonet, "Partitioned block-Gauss-Siedel coupling for dynamic fluid-structure interaction," *Computers and Structures*, vol. 88, pp. 1367–1382, 2010.

[19] C. Farhat, P. Geuzaine, and C. Grandmont, "The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids," *Journal of Computational Physics*, vol. 174, no. 2, pp. 669 – 694, 2001.

[20] D. Boffi and L. Gastaldi, "Stability and geometric conservation laws for ALE formulations," *Computer methods in applied mechanics and engineering*, vol. 193, no. 42, pp. 4717–4739, 2004.

[21] M. L. inne and C. Farhat, "Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations," *Computer Methods in Applied Mechanics and Engineering*, vol. 134, no. 1, pp. 71–90, 1996.

[22] H. Jasak and H. Rusche, "Dynamic mesh handling in OpenFOAM," in *Proceeding of the 47th Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Orlando, FL*, 2009.

[23] H. Jasak and Z. Tukovic, "Dynamic mesh handling in OpenFOAM applied to fluid-structure interaction simulations," in *Proceedings of the V European Conference Computational Fluid Dynamics, Lisbon, Portugal, June*, pp. 14–17, 2010.

[24] A. Sidi, *Methods for Acceleration of Convergence (Extrapolation) of Vector Sequences*, pp. 1828–1846. American Cancer Society, 2009.

[25] D. Young, "Iterative methods for solving partial difference equations of elliptic type," *Transactions of the American Mathematical Society*, vol. 76, no. 1, pp. 92–111, 1954.

[26] R. Mittal and G. Iaccarino, "Immersed boundary methods," *Annual Review of Fluid Mechanics*, vol. 37, no. 1, pp. 239–261, 2005.

[27] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall, "Open MPI: Goals, concept, and design of a next generation MPI implementation," in *Proceedings, 11th European PVM/MPI Users' Group Meeting*, (Budapest, Hungary), pp. 97–104, 2004.

[28] K. Kilroy and M.-S. Corporation, *MSC - Nastran Quick Reference Guide*. MacNeal-Schwendler Corporation, 1998.

[29] A. J. Chorin, "Numerical solution of the navier-stokes equations," *Mathematics of Computation*, vol. 22, no. 104, pp. 745–762, 1968.

[30] J. Smagorinsky, "General circulation experiments with the primitive equations: I. the basic experiment," *Monthly Weather Review*, vol. 91, no. 3, pp. 99 – 164, 1963.

[31] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2 ed., 1992.

[32] L. S. Blackford, A. Petitet, R. Pozo, K. Remington, R. C. Whaley, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, *et al.*, "An updated set of basic linear algebra subprograms (blas)," *ACM Transactions on Mathematical Software*, vol. 28, no. 2, pp. 135–151, 2002.

[33] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*. Philadelphia, PA: Society for Industrial and Applied Mathematics, third ed., 1999.

[34] K.-J. Bathe, "Finite elements in cad and adina," *Nuclear Engineering and Design*, vol. 98, no. 1, pp. 57–67, 1986.

[35] J. A. Swanson, "John swanson and ansys - an engineering success story," in *Proceedings of the Winter Simulation Conference 2014*, pp. 3–4, 2014.

[36] C. A. Truesdell, "The mechanical foundations of elasticity and fluid dynamics," *Indiana University Mathematics Journal*, vol. 1, pp. 125–300, 1952.

[37] "The finite element method: Its basis and fundamentals," in *The Finite Element Method: its Basis and Fundamentals* (O. Zienkiewicz, R. Taylor, and J. Zhu, eds.), p. i, Oxford: Butterworth-Heinemann, seventh ed., 2013.

[38] R. de Borst, M. Crisfield, J. Remmers, and C. Verhoosel, *Nonlinear Finite Element Analysis of Solids and Structures*. Wiley Series in Computational Mechanics, Wiley, 2012.

[39] J. Anderson, J. Wendt, G. Degrez, E. Dick, and R. Grundmann, *Computational Fluid Dynamics: An Introduction*. Springer Berlin Heidelberg, 2013.

[40] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.

[41] T. Sarpkaya, "A critical review of the intrinsic nature of vortex-induced vibrations," *Journal of Fluids and Structures*, vol. 19, pp. 389 – 447, 2004.

[42] N. Rostamy, D. Sumner, D. J. Bergstrom, and J. Bugg, "Instantaneous flow field above the free end of finite-height cylinders and prisms," *Journal of Heat and Fluids Flow*, vol. 43, pp. 120–128, 2013.

[43] C. Norberg, "Flow around rectangular cylinders: Pressure forces and wake frequencies," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 49, pp. 187–196, 1993.

[44] P. Bearman, "Vortex shedding from oscillating bluff bodies," *Ann. Rev. Fluid. Mech.*, vol. 16, pp. 195–222, 1984.

[45] C. Williamson, "Vortex dynamics in the cylinder wake," *Ann. Rev. Fluid. Mech.*, vol. 28, pp. 477–539, 1996.

[46] C. Williamson and A. Roshko, "Vortex formation in the wake of an oscillating cylinder," *Journal of Fluids and Structures*, vol. 2, no. 4, pp. 355–381, 1988.

[47] N. Rostamy, J. F. M. Clean, D. Sumner, D. J. Bergstrom, and J. Bugg, "Flow above free end of a surface-mounted square prism," *the Seventh International Colloquium on Bluff Body Aerodynamics and Applications (BBAA7)*, 2012.

[48] J. F. McClean and D. Sumner, "An experimental investigation of aspect ratio and incidence angle effects for the flow around surface-mounted finite-height square prisms," *Journal Fluids Engineering*, 2013.

[49] H. S. Yoon and J. H. Jung, "Large eddy simulation of flow past a twisted cylinder at a subcritical Reynolds number," *Advances in fluid Mechanics and Heat and Mass Transfer*, no. ISBN:978-1-61804-114-2.

[50] K. Vikestad, J. Vandiver, and C. Larsen, "Added mass and oscillation frequency for a circular cylinder subjected to vortex-induced vibrations amd external disturbances," *Journal of Fluids and Structures*, vol. 14, no. 7, pp. 1071 – 1088, 2000.

[51] C. Williamson and R. Govardhan, "Vortex-induced vibrations," *Ann. Rev. Fluid. Mech.*, vol. 36, pp. 413–455, 2004.

[52] J. Winslow, H. Otsuka, B. Govindarajan, and I. Chopra, "Basic understanding of airfoil characteristics at low Reynolds numbers (104–105)," *Journal of Aircraft*, vol. 55, no. 3, pp. 1050–1061, 2018.

[53] M. Thompson, T.Leweke, and C. Williamson, "The physical mechanism of transition in bluff body wakes," *Journal of Fluids and Structures*, vol. 15, pp. 607–616, 2001.

[54] G. Hou, J. Wang, and A. Layton, "Numerical Methods for Fluid-Structure Interaction - A Review," *Commun. Comput. Phys*, vol. 12, no. 2, pp. 337–377, 2012.

[55] E. Wang, Q. Xiao, and A. Incecik, "Three-dimensional numerical simulation of two-degree-of-freedom VIV of a circular cylinder with varying natural frequency ratios at Re=500," *Journal of Fluids and Structures*, vol. 73, pp. 162 – 182, 2017.

[56] J. Degroote and J. Vierendeels, "Multi-solver algorithms for the partitioned simulation of fluid-structure interaction," *Comput. Methods Appl. Mech. Engrg.*, vol. 200, pp. 2195–2210, 2011.

[57] M. Breuer, D. Nayer, M. Münsch, T. Gallinger, and R. Wüchner, "Fluid-structure interaction using a partitioned semi-implicit predictor-corrector coupling scheme for the application of large-eddy simulation," *Journal of Fluids and Structures*, vol. 29, pp. 107–130, 2012.

[58] D. A. Knoll and D. E. Keyes, "Jacobian-free Newton-Krylov methods: A survey of approaches and applications," *J. Comput. Phys.*, vol. 193, no. 2, p. 357–397, 2004.

[59] D. G. Anderson, "Iterative procedures for nonlinear integral equations," *J. ACM*, vol. 12, no. 4, p. 547–560, 1965.

[60] C. Evans, S. Pollock, L. G. Rebholz, and M. Xiao, "A proof that Anderson acceleration improves the convergence rate in linearly converging fixed point methods (but not in those converging quadratically)," 2018.

[61] F. Xie, J. Deng, Q. Xiao, and Y. Zheng, "A numerical simulation of VIV on a flexible circular cylinder," *Fluid Dynamics Research*, vol. 44, no. 4, p. 045508, 2012.

[62] W. Bangerth, R. Hartmann, and G. Kanschat, "deal.II—a general-purpose object-oriented finite element library," *ACM Transactions on Mathematical Software (TOMS)*, vol. 33, no. 4, p. 24, 2007.

[63] H. Jasak, A. Jemcov, Z. Tukovic, *et al.*, "OpenFOAM: A C++ library for complex physics simulations," in *International workshop on coupled methods in numerical dynamics*, vol. 1000, pp. 1–20, IUC Dubrovnik, Croatia, 2007.

[64] J. Verwer and J. Sanz-Serna, "Convergence of method of lines approximations to partial differential equations," *Computing*, vol. 33, no. 3-4, pp. 297–313, 1984.

[65] F. Lindner, M. Mehl, K. Scheufele, and B. Uekermann, "A comparison of various quasi-Newton schemes for partitioned fluid-structure interaction," in *Proceedings of 6th International Conference on Computational Methods for Coupled Problems in Science and Engineering, Venice*, pp. 1–12, 2015.

[66] C. Michler, S. H. van Brummelen, and R. de Borst, "A monolithic approach to fluid-structure interaction," *Computers & Fluids*, vol. 33, pp. 839–848, 2004.

[67] M. Heil, "An efficient solver for the fully coupled solution of the large-displacement fluid-structure interaction problems," *Comput. Methods Appl. Mech. Engrg.*, vol. 193, pp. 1–23, 2004.

[68] R. Jaiman, X. Jiao, P. Geubelle, and E. Loth, "Conservative load transfer along curved fluid-solid interface with non-matching meshes," *Journal of computational Physics*, vol. 218, pp. 372–397, 2006.

[69] G. Maragkos, S. Verma, A. Trouvé, and B. Merci, "Evaluation of OpenFOAM's discretization schemes used for the convective terms in the context of fire simulations," *Computers & Fluids*, vol. 232, p. 105208, 2022.

[70] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the 1968 23rd ACM national conference*, pp. 517–524, ACM, 1968.

[71] R. van Loon, P. Anderson, F. van de Vosse, and S. Sherwin, "Comparison of various fluid–structure interaction methods for deformable bodies," *Computers*

& Structures, vol. 85, no. 11, pp. 833–843, 2007. Fourth MIT Conference on Computational Fluid and Solid Mechanics.

[72] A. Bavo, G. Rocatello, F. Iannaccone, J. Degroote, J. Vierendeels, and P. Segers, "Fluid-structure interaction simulation of prosthetic aortic valves: Comparison between immersed boundary and arbitrary Lagrangian-Eulerian techniques for the mesh representation," *PloS one*, vol. 11, p. e0154517, 2016.

[73] B. J. Geurts and D. D. Holm, "Commutator errors in large-eddy simulation," *Journal of Physics A: Mathematical and General*, vol. 39, no. 9, p. 2213, 2006.

[74] K. Hughes, R. Vignjevic, J. Campbell, T. D. Vuyst, and N. Djordjevic, "From aerospace to offshore: Bridging the numerical simulation gaps-Simulation advancement for fluid structure interaction problems," *International Journal of Impact Engineering*, vol. 61, pp. 48–63, 2013.

[75] 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 *Fluid Structure Interaction II* (H.-J. Bungartz, M. Mehl, and M. Schäfer, eds.), vol. 73 of *Lecture Notes in Computational Science and Engineering*, pp. 413–424, Springer Berlin Heidelberg, 2010.

[76] J. Degroote, R. Haelterman, S. Annerel, P. Nruggeman, and J. Vierendeels, "Performance of partitioned procedures in fluid-structure interaction," *Computers and Structures*, vol. 88, pp. 446–457, 2010.

[77] F.-B. Tian, H. Dai, H. Luo, J. F. Doyle, and B. Rousseau, "Fluid–structure interaction involving large deformations: 3D simulations and applications to biological systems," *Journal of computational physics*, vol. 258, pp. 451–469, 2014.

[78] C. Wang, B. Ren, and P. Lin, "A coupled flow and beam model for fluid–slender body interaction," *Journal of Fluids and Structures*, vol. 115, p. 103781, 2022.

[79] G. De Nayer, A. Apostolatos, J. Wood, K. Bletzinger, R. Wüchner, and M. Breuer, "Numerical studies on the instantaneous fluid–structure interaction of an air-inflated flexible membrane in turbulent flow," *Journal of Fluids and Structures*, vol. 82, pp. 577–609, 2018.

[80] T. Bano, F. Hegner, M. Heinrich, and R. Schwarze, "Investigation of fluid-structure interaction induced bending for elastic flaps in a cross flow," *Applied Sciences*, vol. 10, no. 18, p. 6177, 2020.

[81] R. W*ü*chner, A. Kpzok, and K.-U. Bletzinger, "Simulation of fluid-structure-interaction with free form membrane structures using an implicit coupling

scheme with adaptive under relaxation," *European Conference on Computational Fluid Dynamics - ECCOMAS CFD 2006*.

[82] K. Bargi, V. Tamimi, and M. Zeinoddini, "VIV of tapered cylinders: 3D LES numerical simulation," *International Journal of Maritime Technology*, vol. 3, no. 0, 2015.

[83] K. Khanafer and R. Bergeur, "Fluid-structure interaction analysis of turbulent pulsative flow within a layered aortic wall as related to aortic dissection," *Journal of Biomechanics*, vol. 42, pp. 2642–2648, 2010.

[84] K. Khanafer, A. Alamiri, and I. Pop, "Fluid-structure interaction analysis of flow and heat transfer characteristics around a flexible microncantilever in a fluid cell," *International Journal of Heat and Mass Transfer*, vol. 53, pp. 1646–1653, 2010.

[85] W. Joppich and M. Kürschner, "Mpcci—a tool for the simulation of coupled applications," *Concurrency and Computation: Practice and Experience*, vol. 18, no. 2, pp. 183–192, 2006.

[86] H.-J. Bungartz, F. Lindner, B. Gatzhammer, M. Mehl, K. Scheufele, A. Shukaev, and B. Uekermann, "precice – a fully parallel library for multi-physics surface coupling," *Computers & Fluids*, vol. 141, pp. 250–258, 2016. Advances in Fluid-Structure Interaction.

[87] J. Vencels, P. Råback, and V. Geža, "EOF-library: Open-source elmer FEM and OpenFOAM coupler for electromagnetics and fluid dynamics," *SoftwareX*, vol. 9, pp. 68–72, 2019.

[88] P. Causin, J. Gerbeau, and F. Nobile, "Added-mass effect in the design of partitioned algorithms for fluid–structure problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 42–44, pp. 4506 – 4527, 2005.

[89] E. H. van Brummelen, "Partitioned iterative solution methods for fluid–structure interaction," *International Journal for Numerical Methods in Fluids*, vol. 65, no. 1-3, pp. 3–27, 2011.

[90] L. Formaggia and F. Nobile, "Stability analysis of second-order time accurate schemes for ALE-FEM," *Computer methods in applied mechanics and engineering*, vol. 193, no. 39, pp. 4097–4116, 2004.

[91] E. Kaya, E. Aulisa, A. Ibragimov, and P. Seshaiyer, "A stability estimate for fluid structure interaction problem with non-linear beam," *Discrete and Continous Dynamical System*, vol. 228, no. 18, pp. 424 – 432, 2009.

[92] I. Barton, "Comparison of simple-and piso-type algorithms for transient flows," *International Journal for numerical methods in fluids*, no. 26, pp. 459–483, 1998.

[93] T. Barth and M. Ohlberger, "Finite volume methods: foundation and analysis," *Encyclopedia of computational mechanics*, 2004.

[94] C. Greenshields and H. Weller, *Notes on Computational Fluid Dynamics: General Principles*. Reading, UK: CFD Direct Ltd, 2022.

[95] J. Smagorinsky, "General circulation experiments with the primitive equations: I. the basic experiment," *Monthly Weather Review*, vol. 91, no. 3, pp. 99 – 164, 1963.

[96] D. K. Lilly, "A proposed modification of the Germano subgrid-scale closure method," *Phys. Fluids*, vol. A, no. 3, pp. 633–635, 1992.

[97] H. Jasak, *(1996) Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows.* PhD thesis, Imperial College London, 1996.

[98] R. Issa, "Solution of the implicitly discretised fluid flow equations by operator-splitting," *Journal of Computational Physics*, vol. 62, no. 1, pp. 40–65, 1986.

[99] A. Yoshizawa and K. Horiuti, "A Statistically-Derived Subgrid-Scale Kinetic Energy Model for the Large-Eddy Simulation of Turbulent Flows," *Journal of the Physical Society of Japan*, vol. 54, no. 8, pp. 2834–2839, 1985.

[100] S. Adhikari, *Damping models for structural vibration*. PhD thesis, University of Cambridge, 2001.

[101] P. Pulay, "Convergence acceleration of iterative sequences. The case of SCF iteration," *Chemical Physics Letters*, vol. 73, no. 2, pp. 393 – 398, 1980.

[102] C. Michler, H. van Brummelen, and R. de Borst, "An investigation of interface-GMRES(r) for fluid–structure interaction problems with flutter and divergence," *J. Computational Mechanics*, vol. 47, pp. 17 – 29, 2011.

[103] P. Wynn, "On the convergence and stability of the epsilon algorithm," *SIAM Journal on Numerical Analysis*, vol. 3, no. 1, pp. 91–122, 1966.

[104] K. Jbilou and H. Sadok, "Vector extrapolation methods. applications and numerical comparison," *Journal of Computational and Applied Mathematics*, vol. 122, no. 1, pp. 149–165, 2000. Numerical Analysis in the 20th Century Vol. II: Interpolation and Extrapolation.

[105] C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Math. Comp.*, vol. 19, pp. 577–593, 1965.

[106] R. Barrett, M. Berry, J. Dongarra, V. Eijkhout, and C. Romine, "Algorithmic bombardment for the iterative solution of linear systems: A poly-iterative approach," *Journal of Computational and Applied Mathematics*, vol. 74, no. 1, pp. 91 – 109, 1996.

[107] H. Jasak, A. Jemcov, and Z. Tukovic, "OpenFOAM: A C++ library for complex physics simulations," in *International Workshop on Coupled Methods in Numerical Dynamics, IUC*, pp. 1–20, 2007.

[108] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

[109] S. Tang, B.-S. Lee, and B. He, "Speedup for multi-level parallel computing," in *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, pp. 537–546, 2012.

[110] C. L. J. Ahrens, B. Geveci, *ParaView: An End-User Tool for Large Data Visualization.* Elsevier, 2005.

[111] D. Sumner, N. Rostamy, D. Bergstrom, and J. Bugg, "Influence of aspect ratio on the mean flow field of a surface-mounted finite-height square prism," *International Journal of Heat and Fluid Flow*, vol. 65, pp. 1–20, 2017.

[112] A. Cesur, C. Carlsson, A. Feymark, L. Fuchs, and J. Revstedt, "Analysis of the wake dynamics of stiff and flexible cantilever beams using POD and DMD," *Computers & Fluids*, vol. 101, pp. 27–41, 2014.

[113] P. Welahettige and K. Vaagsaether, "Comparison of OpenFOAM and ansys fluent," in *Proceedings of the 9th EUROSIM & the 57th SIMS*, (Oulu, Finland), pp. 1005–1012, 2016.

[114] E. Chiriac, D. Broboană, M. Avram, and C. Balan, "Comparative numerical study between OpenFOAM and ansys fluent in a y-junction microchannel," *2019 11th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, pp. 1–4, 2019.

[115] J. S. Ochoa and N. Fueyo, "Large eddy simulation of the flow past a square cylinder," *17th, Efficiency, costs, optimization, simulation and environmental impact of energy on process systems environmental impact of energy on process systems (International conference - ECOS)*, vol. 24, no. 3, pp. 37–50, 2004.

[116] D. Sumner, "Flow above the free end of a surface-mounted finite-height circular cylinder: A review," *Journal of Fluids and Structures*, vol. 43, pp. 41–63, 2013.

[117] A. Saada, *Elasticity: Theory and Applications, Second Edition, Revised & Updated*. J Ross Publishing Series, J. Ross Pub., 2009.

[118] M. Geradin and D. Rixen, *Mechanical Vibrations: Theory and Application to Structural Dynamics*. Wiley, 2015.

[119] A. Chopra, *Dynamics of Structures: Theory and Applications to Earthquake Engineering*. Prentice Hall International Series in Civil Engineering And, Pearson-/Prentice Hall, 2007.

[120] S. Krenk, *Non-Linear Modeling and Analysis of Solids and Structures*. Cambridge University Press, 2009.

[121] K.-J. Bathe and M. M. I. Baig, "On a composite implicit time integration procedure for nonlinear dynamics," *Computers & Structures.*, vol. 83, no. 31-32, pp. 2513–2524, 2005.

[122] D. Mok, W. Wall, and E. Ramm, *Accelerated iterative structuring schemes for stationary fluid-structure interaction*, pp. 1325–1328. 2001.

Appendix A

Finite Volume terminology

A.1 Basic terminology

As described from Eqn (2.12), to obtain a solver to INS, one firstly solves the local problem for each CV in the mesh, then one assembles the local into a global representation of the governing variables. In this assemblage, each integral term becomes a summation of integrated face values and source terms. The governing tensor fields (ϕ_i) are evaluated component-wise, hence dropping the i index.

For each CV there is a cell centroid $(x_i)_P$, and boundary consisting of a set of flat surfaces (facets) with their associated normal n_i, face area S_f and face centroid $(x_i)_f$. The volume of CV is denoted V_P. Each facet is shared with only two CVs, the owner cell P with the neighbour cell N. The vector $(d_i)_{PN}$ defines the vector between the centroid of P and centroid of neightbour N and likewise for the shared facet, $(d_i)_{fN}$ the vector from centroid of P to the sharing facet to the centroid of the neighbour N.

Given any scalar ϕ, a Taylor expansion around the centroid to the first order, leave the integration error second-order accurate,

$$\phi(x) = (\phi)_P + (\phi_{,k})_P(x_k - (x_k)_P) + \mathcal{O}(\|x\|^2). \tag{A.1}$$

For orthogonal mesh, i.e. the facet normal being parallel to the $(d_i)_{PN}$), the divergence term can be estimated by

$$n_j(\phi_{,j})_f = S_f \frac{(\phi_N - \phi_P)}{\|(d_i)_{PN}\|}. \tag{A.2}$$

This is used for the evaluation of the laplacian term in the INS. And for clarity,

$$\int_{CV} \phi dV = V_P \phi_P, \int_{\partial CV} = \sum_f \int_f \phi dA, \int_f \phi dA = S_f \phi_f, \tag{A.3}$$

A.2 Limited advection scheme

The face values ϕ_f (including the gradient above) can be estimated by a "higher-order" scheme, such as the central difference approximation (CD),

$$\phi_{CD} = \gamma \phi_P + (1 - \gamma)\phi_N, \tag{A.4}$$

where $\gamma = \frac{(d_i)_{fN} n_i}{(d_i)_{PN} n_i}$. The mass flux across a given face is defined by $S_i = n_i S_f$,

$$F = S_i(\rho U_i). \tag{A.5}$$

Although being "second-order" accurate, even for non-orthogonal mesh, it introduces a "wiggle" in solving the advection equation, which is commonly applied for FVM in defining schemes to compute the face values. The upwind scheme resolves this,

$$\phi_f = \phi_P \ , \ F \geq 0, \phi_f = \phi_N \ , \ F < 0, \tag{A.6}$$

but lowers the accuracy to first-order and is phrased as ϕ_{UD}. Normally one applies a limiter (Ψ) between those,

$$\phi_f = \Psi \phi_{CD} + (1 - \Psi)\phi_{UD}. \tag{A.7}$$

Along $(d_i)_{PN}$ and for given flow direction, define N as U while upwind and D downwind with respect to source value P, then setting $r = \frac{\phi_P - \phi_U}{\phi_D - \phi_P}$, the ratio of the gradient of face values between upwind/downwind cell, one obtains a parameter to which one can defines schemes to compute the face values.

A.3 TVD scheme

A TVD scheme is an "entropy" satisfying scheme such that for any update (i.e. $\phi^n \rightarrow \phi^{n+1}$), with summation of fluxes (TV),

$$TV(\phi^n) = \sum_f \phi_N^n - \phi_P^n, \tag{A.8}$$

it is not increasing, hence satisfying $TV(\phi^{n+1}) \leq TV(\phi^n)$. A limiter defined by,

$$\Psi(r) = max(min(\frac{2r}{\kappa}, 1), 0), \tag{A.9}$$

is a TVD scheme. This provides a "wiggle" free solution but introduces numerical diffusion. The κ is the "blend" between 0 and 1, where 0 mean $\Phi = 1$ and vice versa $\Phi = 0$. The Eqn (A.7) is applied in OpenFOAM with the keyword "limitedLinear" while Eqn (A.4) by "linear".

Appendix B

Finite Element terminology

Voigt notation introduces matrices that is used in the implementation of the solid solver.

B.1 Space discretisation, the shape matrix: H_{ij}

The Finite Element Method approximates the continuous displacement field to all degrees of freedom with respect to an element vector a_i^e of length $3 \times n$ using the vertexes of the polyhedron (nodes) as discrete representation.

$$a(\xi, \eta, \zeta)_i = \sum_{ik} h(\xi, \eta, \zeta)_i^k a_{3k+i-3}^e, \tag{B.1}$$

where h_i^k are the shape functions, normally in terms of the local coordinate system using iso-parametric coordinates, which introduces the local shape function matrix $H_{ij}(\xi, \eta, \zeta)$ matrix,

$$H_{ij} = \begin{pmatrix} h_1^1 & 0 & 0 & h_1^2 & 0 & 0 & \dots & h_1^n & 0 & 0 \\ 0 & h_2^1 & 0 & 0 & h_2^2 & 0 & \dots & 0 & h_2^n & 0 \\ 0 & 0 & h_3^1 & 0 & 0 & h_3^2 & \dots & 0 & 0 & h_3^n \end{pmatrix}. \tag{B.2}$$

B.2 Equation of motion: operator matrix L_{ij}

Operator matrix for σ_j in Eqn (4.19),

$$L_{ij} = \begin{pmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{pmatrix}. \tag{B.3}$$

B.3 Virtual work equation: operator matrix L_{ij}

L_{ij} operator matrix for stress-strain displacement becomes for the virtual work equation for finite strain,

$$L_{ij} = \begin{bmatrix} F_{XX}\frac{\partial}{\partial X} & F_{YX}\frac{\partial}{\partial X} & F_{ZX}\frac{\partial}{\partial X} \\ F_{XY}\frac{\partial}{\partial Y} & F_{YY}\frac{\partial}{\partial Y} & F_{ZY}\frac{\partial}{\partial Y} \\ F_{XZ}\frac{\partial}{\partial Z} & F_{YZ}\frac{\partial}{\partial Z} & F_{ZZ}\frac{\partial}{\partial Z} \\ F_{XX}\frac{\partial}{\partial Y} + F_{YX}\frac{\partial}{\partial X} & F_{YX}\frac{\partial}{\partial Y} + F_{YY}\frac{\partial}{\partial X} & F_{ZX}\frac{\partial}{\partial Y} + F_{ZY}\frac{\partial}{\partial X} \\ F_{XY}\frac{\partial}{\partial Z} + F_{XZ}\frac{\partial}{\partial Y} & F_{YY}\frac{\partial}{\partial Z} + F_{YZ}\frac{\partial}{\partial Y} & F_{ZY}\frac{\partial}{\partial Z} + F_{ZZ}\frac{\partial}{\partial Y} \\ F_{XZ}\frac{\partial}{\partial X} + F_{XX}\frac{\partial}{\partial Z} & F_{YZ}\frac{\partial}{\partial X} + F_{YX}\frac{\partial}{\partial Z} & F_{ZZ}\frac{\partial}{\partial X} + F_{ZX}\frac{\partial}{\partial Z} \end{bmatrix}, \tag{B.4}$$

where (X_i) is reference coordinate system and (ξ_i) the current coordinate F_{ij} the finite strain matrix, used to connecting the stress/strain between current and reference, see Figure B.1. The local coordinate is related to the global by mapping $x = x(\zeta)$ and

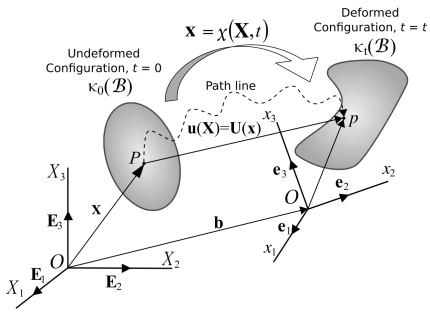

Figure B.1: The large deformation problem, $X \to x\ (\xi, \eta, \zeta)$ [Wikimedia Commons search: Sanpaz]

the integrand interrelates $dV = dxdydz = (detF_{ij})d\xi d\eta d\zeta$, where F is the jacobian matrix,

$$F_{ij} = \frac{\partial x_i}{\partial \xi_j} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}, \tag{B.5}$$

called the deformation matrix. Define the displacement vector a accordingly $x_i = \xi_i + a_i$,

$$F_{ij} = \delta_{ij} + \frac{\partial a_i}{\partial \xi_j}. \tag{B.6}$$

This relates to Green-Lagrange strain tensor γ_{ij},

$$dx_i dx_i - d\xi_i d\xi_i = d\xi_i \left[F_{ki}F_{kj} - \delta_{ij}\right] d\xi_j = 2d\xi_i \gamma_{ij} d\xi_j. \tag{B.7}$$

From polar decomposition theorem $F = RU = VR$, involving a pair of stretch and rotation tensors, gives Cauchy-Green deformation tensor $C_{ij} = F_{ki}F_{kj} \equiv U^2$,

Cauchy-Green deformation tensor $B_{ij} = F_{ik}F_{jk} \equiv V^2$, see Figure B.2. Cauchy stress tensor relates to the current configuration; the corresponding tensor with respect to reference is called the Second Piola-Kirchhoff stress tensor τ_{ij},

$$\sigma_{ij} = \frac{1}{detF} F_{ik}\tau_{kl}F_{jl}. \tag{B.8}$$

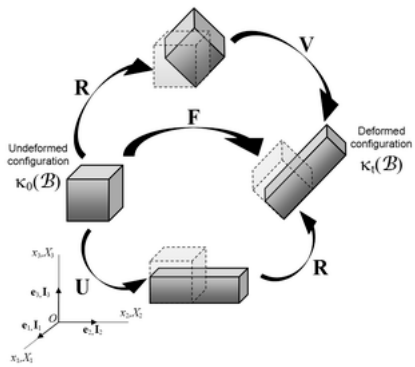

Figure B.2: The transformation: pullback/pushforward. [Wikimedia Commons search: Sanpaz]

B.4 Non-linear K_{NL}: T,B_{NL}

The geometric stiffness tensor use the following matrix that collects the stresses for the Second Picola-Kirchhoff Stress tensor,

$$\mathrm{T} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} & 0 & 0 & 0 & 0 & 0 & 0 \\ \tau_{xy} & \tau_{yy} & \tau_{yz} & 0 & 0 & 0 & 0 & 0 & 0 \\ \tau_{zx} & \tau_{yz} & \tau_{zz} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tau_{xx} & \tau_{xy} & \tau_{xz} & 0 & 0 & 0 \\ 0 & 0 & 0 & \tau_{xy} & \tau_{yy} & \tau_{yz} & 0 & 0 & 0 \\ 0 & 0 & 0 & \tau_{zx} & \tau_{yz} & \tau_{zz} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \tau_{xx} & \tau_{xy} & \tau_{xz} \\ 0 & 0 & 0 & 0 & 0 & 0 & \tau_{xy} & \tau_{yy} & \tau_{yz} \\ 0 & 0 & 0 & 0 & 0 & 0 & \tau_{zx} & \tau_{yz} & \tau_{zz} \end{bmatrix}, \tag{B.9}$$

together with the following matrix with element derivatives used to compute the green strain-displacement,

$$B_{NL} = \begin{bmatrix} \frac{\partial h_1}{\partial X} & 0 & 0 & \dots & \frac{\partial h_n}{\partial X} & 0 & 0 \\ \frac{\partial h_1}{\partial Y} & 0 & 0 & \dots & \frac{\partial h_n}{\partial Y} & 0 & 0 \\ \frac{\partial h_1}{\partial Z} & 0 & 0 & \dots & \frac{\partial h_n}{\partial Z} & 0 & 0 \\ 0 & \frac{\partial h_1}{\partial X} & 0 & \dots & 0 & \frac{\partial h_n}{\partial X} & 0 \\ 0 & \frac{\partial h_1}{\partial Y} & 0 & \dots & 0 & \frac{\partial h_n}{\partial Y} & 0 \\ 0 & \frac{\partial h_1}{\partial Z} & 0 & \dots & 0 & \frac{\partial h_n}{\partial Z} & 0 \\ 0 & 0 & \frac{\partial h_1}{\partial X} & \dots & 0 & 0 & \frac{\partial h_n}{\partial X} \\ 0 & 0 & \frac{\partial h_1}{\partial Y} & \dots & 0 & 0 & \frac{\partial h_n}{\partial Y} \\ 0 & 0 & \frac{\partial h_1}{\partial Z} & \dots & 0 & 0 & \frac{\partial h_n}{\partial Z} \end{bmatrix}. \tag{B.10}$$

Scientific publications

Author contributions

Paper I: A numerical study of partitioned fluid-structure interaction applied to a cantilever in incompressible turbulent flow.

I formulated the research goals with coauthor, creating the setup of all cases, coded the solution procedure, executed the simulations, preprocessing and postprocessing. The results were analysed in cooperation with the coauthor. I am the main author of the manuscript.

Paper II: On stability and relaxation techniques for partitioned fluid-structure interaction simulations

I formulated the research goals with coathor, creating the setup of all cases, executing the simulations, preprocessing and postprocessing. The results were analysed in cooperation with the coauthor. I am the main author of the manuscript.

Paper III: The influence of hydrodynamic damping on the motion of an immersed elastic cantilever

I formulated the research goals together with coauthors, creating the setup of all cases, executed the simulations, preprocessing and postprocessing. The results were analysed in cooperation with the coauthors. I am the main author of the manuscript.

Paper IV: A study of efficiency for partitioned FSI with application to cantilever in transient regime with turbulence modelling

I formulated the research goals, creating the setup of all cases, implemented additional relaxation methods, executed the simulations, preprocessing and postprocessing. I am the sole author of the manuscript.